Advant® OCS

AdvaInform® History

User’s Guide

ABB

Use of **DANGER**, **WARNING**, **CAUTION**, and **NOTE**

This publication includes, **DANGER**, **WARNING**, **CAUTION**, and **NOTE** information where appropriate to point out safety related or other important information.

DANGER	Hazards which could result in severe personal injury or death
WARNING	Hazards which could result in personal injury
CAUTION	Hazards which could result in equipment or property damage
NOTE	Alerts user to pertinent facts and conditions

Although **DANGER** and **WARNING** hazards are related to personal injury, and **CAUTION** hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all **DANGER**, **WARNING**, and **CAUTION** notices.

TRADEMARKS

Advant, AdvaInform and AdvaBuild are registered trademarks of ABB Industrial Systems Inc.

MOD 300, Batch 300, and Taylor are Trademarks of ABB Industrial Systems Inc.

ORACLE, SQL*Plus are registered trademarks of Oracle Corporation.

NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB Industrial Systems Inc. ABB Industrial Systems Inc. assumes no responsibility for any errors that may appear in this document.

In no event shall ABB Industrial Systems Inc. be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB Industrial Systems Inc. be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without ABB Industrial Systems Inc.'s written permission, and the contents thereof must not be imparted to a third party nor be used for any unauthorized purpose.

The software described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

Copyright © 1997 ABB Industrial Systems Inc. All rights reserved.

TABLE OF CONTENTS

Chapter 1 - Introduction

1.1 General Information........ 1-1
1.2 Equipment Requirements........ 1-2
1.3 Manual Organization 1-2
  1.3.1 Differences Between Master and MOD 300........ 1-4
1.4 Conventions 1-5
1.5 Related Documentation........ 1-6
1.6 Release History 1-7
  1.6.1 Release 1.0........ 1-7
  1.6.2 Release 1.1........ 1-7
  1.6.3 Release 1.2........ 1-7
  1.6.4 History 2.1 1-8
  1.6.5 History 2.1/1 (MOD 300 Only) 1-8
  1.6.6 History 2.2 1-9
1.7 Terminology........ 1-9
1.8 Product Overview 1-12
  1.8.1 Historical Logs........ 1-13
    1.8.1.1 Log Hierarchy - Composite Logs........ 1-14
    1.8.1.2 Log Groups 1-17
    1.8.1.3 Log Sets........ 1-17
    1.8.1.4 Log Typicals........ 1-17
    1.8.1.5 Composite Log Typicals 1-17
    1.8.1.6 Log Types........ 1-18
    1.8.1.7 Log Attributes 1-18
    1.8.1.8 Log Entries........ 1-19
  1.8.2 History Database Configuration 1-19
  1.8.3 Data Collection 1-19
  1.8.4 Data Storage........ 1-20
    1.8.4.1 File Storage vs. Oracle Tables........ 1-20
    1.8.4.2 Log Capacity 1-20
  1.8.5 Data Retrieval 1-20
    1.8.5.1 Data Presentation 1-21
    1.8.5.2 Other Applications that use Historical Data 1-22
  1.8.6 Data Archival and Restore........ 1-24
  1.8.7 History Status........ 1-25
1.9 User Interface........ 1-25

CONTENTS (continued)

Chapter 2 - Installation

2.1 Site Planning Environment 2-1
  2.1.1 Guidelines for planning installation of Advant Station........ 2-1
2.2 Setup........ 2-1
  2.2.1 External Display Access (MOD 300) 2-1
2.3 Shut-down Procedures 2-1
2.4 Start-up Procedures 2-1
2.5 Product Verification 2-2

Chapter 3 - Configuration and Application Building

3.1 Design Considerations 3-2
  3.1.1 Recommended Method for History Database Configuration........ 3-2
  3.1.2 Data Sources for Numeric Logs........ 3-2
  3.1.3 Data Collection Considerations........ 3-3
    3.1.3.1 Collection Mode and Collection Type........ 3-3
    3.1.3.2 Sample Interval........ 3-5
    3.1.3.3 Calculation Algorithm and Storage Interval........ 3-5
    3.1.3.4 Blocking Rate 3-6
    3.1.3.5 Activation 3-6
    3.1.3.6 Log Period 3-6
    3.1.3.7 Compaction Deadband & Deadband Storage Interval 3-7
    3.1.3.8 Data Collection Examples 3-10
  3.1.4 File Storage vs. Oracle Tables........ 3-16
    3.1.4.1 Comparison of TYPE1 and TYPE2 File-based Logs..... 3-17
    3.1.4.2 Disk Requirements for Oracle-based Logs........ 3-20
    3.1.4.3 Disk Requirements for File-based Numeric Logs........ 3-20
  3.1.5 Presentation (Trend), Engineering Units, and Alarm Attributes........ 3-20
    3.1.5.1 Presentation Units........ 3-20
    3.1.5.2 Engineering Units, and Alarm Attributes........ 3-21
    3.1.5.3 Configuring the Presentation, Engineering Units and Alarm Attributes........ 3-21
  3.1.6 Data Retrieval Considerations........ 3-22
    3.1.6.1 When to Use Log Name or Access Name 3-22
    3.1.6.2 Retrieval for AdvaCommand Trend Displays in MOD 300........ 3-23
    3.1.6.3 Retrieval for AdvaCommand Trend Displays in Master 3-25
    3.1.6.4 Retrieval for SQL*Connect and AdvaInform Reports... 3-26
    3.1.6.5 Retrieval for AdvaInform User API 3-28

CONTENTS (continued)

3.1.6.6 Criteria for selection for Seamless Retrieval 3-29
3.1.6.7 Interpolation 3-30
3.1.7 Archive Considerations.. 3-31
3.1.8 History Log Attributes 3-31
3.2 Capacity and Performance 3-39
3.2.1 CPU Performance Limitations.. 3-39
3.2.1.1 Fastest Sample Rate 3-39
3.2.1.2 Samples per Second 3-39
3.2.1.3 Data Storage Per Second.. 3-39
3.2.1.4 Maximum Log Size... 3-40
3.2.1.5 Log Group, Log Set, and Composite Log Size 3-40
3.2.1.6 Number of Primary Logs 3-40
3.2.1.7 Number of Log Objects.. 3-40
3.2.2 Disk Capacity... 3-40
3.2.2.1 Default Oracle Tablespace 3-40
3.2.3 Compatibility 3-41
3.3 Application Start-up.. 3-42
3.3.1 Accessing History on Advant Station w/ AdvaCommand (MOD 300) 3-42
3.3.2 Accessing History on an Advant Station w/AdvaInform 3-42
3.4 Tutorial.. 3-43
3.4.1 How to Begin a History Builder Session... 3-44
3.4.2 Tutorial 1 - Storing Numeric Values at a Periodic Rate....................... 3-45
3.4.2.1 Configure a Log Group 3-46
3.4.2.2 Build a Basic Composite Log 3-48
3.4.2.3 Edit Log Attributes... 3-51
3.4.3 Tutorial 2 - Building Log Hierarchies Using a Composite Log Typical3-54
3.4.3.1 Build Log Typicals... 3-54
3.4.3.2 Build Log Groups.. 3-57
3.4.3.3 Build a Composite Log Typical 3-59
3.4.3.4 Build Composite log using a Composite Log Typical ... 3-63
3.5 Application Procedures... 3-65
3.5.1 Getting Started 3-65
3.5.2 Create History Directories 3-65
3.5.3 Log Sets 3-65
3.5.3.1 How to Assign Logs to a Log Set 3-65
3.5.4 A Simple Log without using Typicals 3-67
3.5.5 Simple Logs using a Log Typical 3-67
3.5.6 Composite Logs using a Log Typical and Composite Log Typical...... 3-67

CONTENTS (continued)

3.5.7 Configure History Node to Support Archive Function 3-67
3.5.8 How to Build Report Logs 3-68
3.5.9 How to Build Message Logs for MOD 300 3-69
3.5.9.1 Advant OCS Database Requirements for History Message Logs 3-69
3.5.9.2 Guidelines for Configuring Message Logs for MOD 300 3-70
3.5.10 Configure Logs to be Archived with Log Typicals 3-70
3.5.11 How to Edit Log Attributes 3-71
3.5.12 Error Messages 3-73
3.5.13 How to Configure History Logs in a Text File 3-74
3.5.13.1 Procedure for Configuring History Database with Text Files 3-75
3.5.13.2 Dual History Example (MOD 300 Only) 3-86
3.5.13.3 dbDump and dbLoad Usage 3-89
3.6 Configuration/Application Menus 3-91
3.6.1 Common AdvaInform History Menus 3-91
3.6.1.1 File Menu 3-91
3.6.1.2 Help Menu 3-93
3.6.2 History Builder Window 3-94
3.6.2.1 File Menu 3-95
3.6.2.2 Tools Menu 3-95
3.6.3 History Configuration Window 3-96
3.6.3.1 History Manager Overview 3-98
3.6.3.2 Menus for the History Configuration Window 3-98
3.6.3.3 File Menu 3-99
3.6.3.4 Tools Menu 3-99
3.6.3.5 Utility Menu (MOD 300) 3-101
3.6.3.6 Help Menu 3-102
3.6.4 Composite Log Typical Window 3-103
3.6.4.1 File Menu 3-105
3.6.4.2 Edit Menu 3-105
3.6.4.3 Help Menu 3-106
3.6.5 Log Typical Window 3-106
3.6.6 Log Group Window 3-107
3.6.6.1 File Menu 3-109
3.6.6.2 Action Menu 3-109
3.6.6.3 Help Menu 3-109

CONTENTS (continued)

3.6.7 Log Set Window 3-110
3.6.7.1 File Menu 3-111
3.6.7.2 Action Menu 3-111
3.6.7.3 Help Menu....................3-111
3.6.8 Composite Log Window 3-112
3.6.8.1 File Menu 3-113
3.6.8.2 Log Menu 3-114
3.6.9 Archive Device Window....................3-117
3.6.9.1 File Menu 3-118
3.6.9.2 Action Menu 3-118
3.6.9.3 Help Menu....................3-118
3.6.10 Archive Group Window....................3-118
3.6.10.1 File Menu 3-121
3.6.10.2 Action Menu 3-121
3.6.10.3 Help Menu.................... 3-121
3.6.11 Directory Maintenance Window.................... 3-122
3.6.11.1 Guidelines for Configuring File-based Storage 3-122
3.6.11.2 Description of Directory Maintenance Window 3-124
3.6.11.3 Add Button 3-125
3.6.11.4 Delete Button 3-127
3.6.12 PDL Archive Configuration Window (MOD 300).................... 3-128

Chapter 4 - Runtime Operation

4.1 Product Operation 4-1
4.2 Operating Overview.................... 4-1
4.3 Runtime Tutorial 4-1
4.4 Operating Instructions.................... 4-1
4.4.1 History Status.................... 4-2
4.4.1.1 How to Display Detailed Runtime History Status Information.................... 4-3
4.4.1.2 How to Build a Custom List 4-5
4.4.1.3 How to View a Custom List.................... 4-7
4.4.1.4 How to View Detailed History Status Information for a Log Set.................... 4-8
4.4.1.5 How to Display a Filtered Log List 4-8
4.4.1.6 How to Activate and Deactivate Logs via the Detail Window 4-9
4.4.1.7 How to Change Which Attributes are Displayed in the Detail Window 4-10

CONTENTS (continued)

4.4.1.8 How to Configure Page Size 4-11
4.4.2 Starting and Stopping Data Collection.. 4-11
4.4.2.1 Activating and De-activating a Log Set 4-12
4.4.2.2 Activating and De-activating a Log Group 4-14
4.4.2.3 Activating and De-activating an Individual Log 4-17
4.4.3 Archiving Data for Logs 4-19
4.4.3.1 How to Initialize the Media... 4-19
4.4.3.2 How to Archive Logs 4-21
4.4.3.3 How to View Archive Database Information 4-27
4.4.3.4 How to Restore Archive Logs from Archive Media to Restored History Database 4-30
4.4.3.5 How to Delete Archive Entries from the Archive Database 4-31
4.4.3.6 How to Delete Restored Logs.. 4-31
4.4.3.7 How to Add Database Entries from Archive Media to History Database 4-33
4.4.3.8 How to Use the Filter and Search Dialogs 4-35
4.4.4 Handling Production Data Logs (MOD 300)....................................... 4-38
4.4.4.1 How to Display Jobs and Batches 4-38
4.4.4.2 How to Delete a PDL and/or Batch Task 4-39
4.5 Runtime Menus 4-40
4.5.1 Action Menu in Log Set and Log Group Windows 4-40
4.5.2 Menus for Initialize Archive Media Window 4-41
4.5.3 Menus for the View Logs Window 4-42
4.5.3.1 View Logs Window - File Menu 4-42
4.5.3.2 View Logs Window - View Menu 4-43
4.5.3.3 View Logs Window - Select Menu................................ 4-44
4.5.4 Menus for View Archive Media Window 4-45
4.5.4.1 View Archive Media Window - File Menu 4-45
4.5.4.2 View Archive Media Window - View Menu................. 4-47
4.5.4.3 View Archive Media Window - Select Menu 4-51
4.5.4.4 View Archive Database Window 4-51
4.5.4.5 View Archive Database - File Menu 4-51
4.5.4.6 Archive Database Window - View Menu....................... 4-53
4.5.4.7 Archive Database Window - Select Menu 4-58
4.5.5 Menus for the PDL Window (MOD 300) 4-58
4.5.5.1 PDL Window - File menu 4-58
4.5.6 Menus for the Runtime History Status Window 4-58
4.5.6.1 Runtime History Status, Detail Display - File menu...... 4-58

CONTENTS (continued)

4.5.6.2 Runtime History Status, Detail Display - Tools menu... 4-59
4.5.6.3 Runtime History Status Detail Display - Actions Menu 4-60

Chapter 5 - Maintenance

5.1 Preventive Maintenance............ 5-1
5.1.1 Problem Reporting............ 5-1
5.1.2 The HP-UX kernel............ 5-1
5.2 Hardware Indicators............ 5-1
5.3 Error Messages for Systems with Master Software............ 5-2
5.3.1 Oracle *oerr* on-line tool 5-2
5.3.2 Messages Box 5-3
5.3.3 History Diagnostic Messages............ 5-4
5.3.3.1 Main History 5-4
5.4 Error Messages for Systems with MOD 300 Software............ 5-5
5.4.1 Accessing Error Messages 5-5
5.4.2 Error Message Format 5-7
5.4.3 History Diagnostic Messages............ 5-7
5.4.3.1 AP 81 - Main History............ 5-7
5.4.3.2 AP 82 - Log Manager............ 5-12
5.4.3.3 AP 83 - History Builder 5-13
5.5 Fault Finding and Repair 5-14
5.6 Database Maintenance Functions 5-16
5.6.1 How to Access hsDBMaint Functions............ 5-17
5.6.2 Exit hsDBMaint 5-19
5.6.3 Database Free Space Report 5-19
5.6.4 Entry Tables Report 5-20
5.6.4.1 Running Entry Tables Report Function from Command Line............ 5-20
5.6.4.2 Running Entry Tables Report Function from the Menu 5-20
5.6.5 Update Deadband Ratio On-line............ 5-22
5.6.6 Create or Drop a Product Database............ 5-23
5.6.7 How to Extend Tablespace for History Files 5-24
5.6.7.1 Guidelines for Extending Tablespace............ 5-24
5.6.7.2 Procedure for Extending Tablespace............ 5-24

CONTENTS (continued)

5.6.8 Add, Alter, Delete, or List Numeric Log Directories............................ 5-26
  5.6.8.1 Add a Directory 5-27
  5.6.8.2 Update Directory List.. 5-27
  5.6.8.3 Delete From Directory List 5-27
  5.6.8.4 List the Directory.. 5-28
5.6.9 Reset Object Status States.. 5-29
5.6.10 Cascade Attributes for Composite Logs 5-30
5.6.11 Gather CPU Performance Statistics 5-31
5.6.12 Clean History Database... 5-33
5.6.13 Restore or initialize History Logs 5-33
5.6.14 Stagger Collection and Storage... 5-34
5.6.15 Purge History Data.. 5-38
5.7 Backup and Restore.. 5-40

Appendix A - History Resource Considerations

A.1 Resource Requirements Related to History Configuration..................................... A-1
A.2 Resource Requirements Related to User Applications A-2
A.3 Other Considerations... A-2
A.4 Determining Resource Usage for Your Application A-2
A.5 Temporary File Disk Space for History A-5

INDEX

Chapter 1 Introduction

1.1 General Information

This book describes AdvaInform® History software (hereafter referred to as History or the History package). This book describes AdvaInform History for both Advant® OCS with Master software and with MOD 300 software. There are some differences in the application and operation of the History package between Master and MOD 300. These are mainly related to differences in the object types between the two systems. Wherever such differences occur in the book they will be marked with a parenthesis (Master) or (MOD 300). A detailed description of the differences is given in Section 1.3.1, Differences Between Master and MOD 300.

History is an optional package runs on an Advant Station with AdvaInform software. In systems with MOD 300 software, the History package can also run on an Advant Station with AdvaCommand® software. The operation of the History package is the same whether you run it on an Advant Station with AdvaInform or AdvaCommand software. The instructions in this book are applicable for both cases. The only difference is how you access History. Details are provided in Section 3.3, Application Start-up. Also, History on the Advant Station with AdvaInform software will support a larger History database than History on the Advant Station with AdvaCommand software.

History supports collection, storage, archival, and retrieval of process data. Other packages can access historical data for use in reports and graphic displays, and as a basis for statistical process control. The Advant Station with AdvaInform software can display historical data in chart form. Historical information can also be displayed on AdvaCommand® trend graphs, and on AdvaInform Display client nodes.

The History package includes the History Builder for configuring the historical database and manipulating archived data, and the History Status function for monitoring history logs. You can operate the History Builder and History Status via the IMS menu or via windows opened on either Advant Stations with AdvaCommand software or X Terminals.

This book is intended for application engineers who configure the historical database, and operators and plant supervisors who view, interpret and respond to historical data during runtime. Operators may redefine history objects during runtime if given the proper authority.

As a prerequisite, you should understand distributed automated process control, the hardware and software functionality of an Advant Open Control System (OCS), the platform and operating system that History is operating on, and information storage in databases.

This book is not intended to be the sole source of instruction for AdvaInform History software. It is recommended that those people who use AdvaInform History attend the applicable training courses offered by ABB.

1.2 Equipment Requirements

The hardware platform for History is the Advant Station 500 Series. The Advant Station is equipped with the hardware and software required to support the software packages that you order. The memory provided to support basic functions in these stations, and the memory requirements for various options are described in the *Advant Station 500 Series IMS User's Guide*, and the *Advant Station Operator Station User's Guide*.

Other software packages that either use or support AdvaInform History functions are:

- AdvaInform Basic Functions - Supports on screen viewing of historical data
- AdvaCommand Software - Displays trend graphs of process variables. These graphs can use historical information
- AdvaBuild Environment Builder - As part of environment configuration, you can select which process variables to include in AdvaCommand trend displays
- AdvaBuild Display Builder - Supports trend graphs on custom built displays
- AdvaInform Reports - Supports viewing and printing of reports which can contain historical data
- AdvaInform Display - Display historical data in graphical formats, including trends
- AdvaInform SPC (Statistical Process Control) - Analyzes historical data to determine if a process is producing consistent results.
- AdvaInform PDL (Production Data Log) - Supports data logging and retrieval for batch applications

NOTE

SPC and PDL are options for systems with MOD 300 software only.

1.3 Manual Organization

This book is organized in chapters and sections as shown in Figure 1-1. This is the general organization for all Advant OCS user's guides. Some chapters and sections may not apply to AdvaInform History.

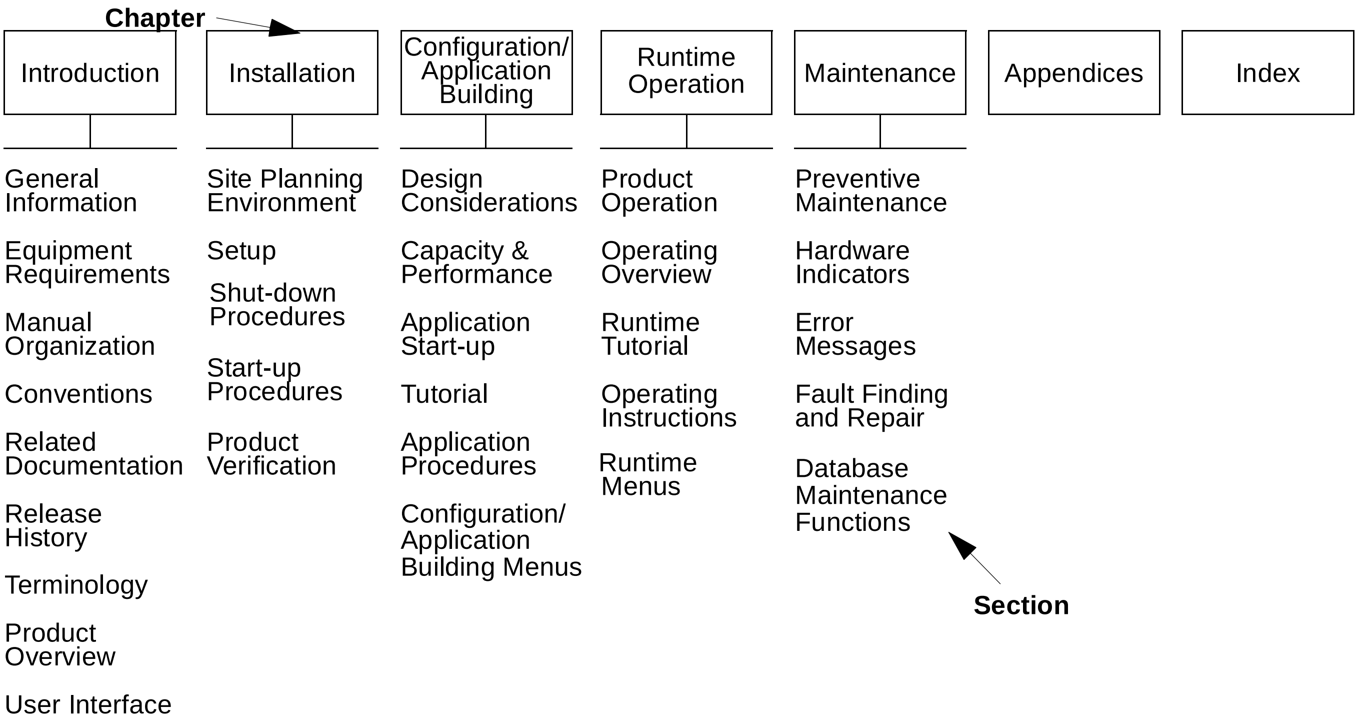


Figure 1-1. Advant Technology Documentation Standardized Format

How to Use this Book

This book covers both history configuration and runtime operation. Read Section 1.8, Product Overview to get a general understanding of the configuration and runtime functions.

Refer to Chapter 3, Configuration and Application Building for instructions on how to use the History Builder to configure your History database. This chapter is organized so you can learn how to use the History Builder without having to read the entire book. First, refer to the tutorial in Section 3.4 to get some quick “hands-on” experience. If you need further information regarding a procedure that is not covered in the tutorial, refer to Section 3.5, Application Procedures. This section provides step-by-step instructions for implementing common history applications. Section 3.6, Configuration/Application Menus describes each History Builder window and associated dialogs and menus.

Section 3.1, Design Considerations describes the object attributes such as storage interval and data compaction that you configure to meet the functional requirements of your application. You should read this section before you begin configuring your history database. Refer back to Section 3.1, Design Considerations as needed when you actually do the configuration.

Chapter 4, Runtime Operation provides instructions for starting and stopping data collection for history logs during runtime. It also provides instructions for using the archive function.

Chapter 5, Maintenance describes error messages applicable to this software package. It also provides instructions for database maintenance functions such as extending tablespace.

1.3.1 Differences Between Master and MOD 300

The following is a brief summary of the main differences in the application and operation of AdvaInform History for Master and MOD 300.

History on an Advant Station with AdvaCommand Software

In systems with MOD 300 software, AdvaInform History can run on an Advant Station with either AdvaInform software or AdvaCommand software. In systems with Master software, AdvaInform History runs on an Advant Station with AdvaInform software. All references to History on an Advant Station with AdvaCommand software are for MOD 300 applications only.

Object Types

While many AdvaInform object types are common for MOD 300 and Master, there are also object types that are unique to each. Refer to the applicable Object Type Reference Manual for details. There is a separate manual for MOD 300 object types and Master object types.

MEASURE and VALUE Object Attributes

MEASURE is an object attribute for historical logging in MOD 300 applications. The equivalent attribute for Master applications is VALUE. These attributes are frequently used in examples in this book. In some cases, example for both MEASURE and VALUE are provided. In other cases just one attribute or the other is shown. In all cases, the examples can be made applicable for both Master and MOD 300 by simply substituting the correct attribute.

Event and Alarm Handling

In Master, events, alarms and alarm acknowledge is supported. In MOD 300, TCL Billboard Messages can be created. In Master the messages are also stored locally which is not the case for MOD 300. The term event and alarm messages is used throughout the document even though only part of the functionality is applicable for MOD 300. Event examples and references in this book are only applicable for Master.

TCL, CCF, PDL and SPC

TCL and CCF are software packages for MOD 300 applications. All examples and references for TCL and CCF are only applicable for MOD 300. SPC and PDL are options to History for MOD 300 applications. These options are not available for systems with Master software.

TTD Logs

TTD refers to Time Tagged Data. This is the primary logging feature in Master-based Advant Controllers. TTD logs can be stored in History. TTD examples and references are only applicable for Master.

Dual History

The dual history feature is only applicable for MOD 300.

1.4 Conventions

The following conventions are used throughout this manual for the presentation of material:

- The words in proper names of screen elements (for example, the title in the title bar of a window, the label for a field of a dialog box) are initially capitalized.
- The names of keyboard keys are boldfaced.
    - Initial capital letters are used for the name of a keyboard key if it is labeled on the keyboard. For example, the **Return** key or **<Return>**.
    - Lowercase letters are used for the name of a keyboard key that is not labeled on the keyboard. For example, the **space bar**, **comma key**, and so on.
    - In descriptions, the word "key" is used in conjunction with its name. For example, pressing the **Return** key initiates the process.
    - In tables and procedures, angle brackets replace the word "key." For example, press **<Return>**.
    - A dash is used to indicate that you must simultaneously press several keys. For example, **<Ctrl-C>** indicates that you must hold down the **Control** key while pressing the **C** key.
    - The phrase "press and release" is used to indicate that you sequentially press several keys.
- The names of push and toggle buttons are boldfaced.
    - Initial capital letters are used for the name of a push or toggle button labeled on the display. For example, the **OK** button.
    - In descriptions, tables and procedures, the word "button" is used in conjunction with its name. For example, pressing the **OK** button initiates the process.
- The names of menus and menu items are boldfaced.
    - Initial capital letters are used for the name of menus listed in the menu bar. For example, the **File** menu.
    - Initial capital letters are used for the name of menu items. For example, to open a new window, choose **New** from the **File** menu.
- System prompts/messages are shown in the Courier font, and user responses/input are in the boldfaced Courier font. For example, if you enter a value out of range, the following message is displayed:

    ```
    Entered value is not valid. The value must be 0-30.
    ```

    In a tutorial, you may be told to enter the string TIC132 in a field. The string is shown as follows in the procedure: **TIC132**

1.5 Related Documentation

Table 1-1 lists the documentation for hardware and software associated with AdvaInform History. Not all documents are applicable, depending on whether you use History for MOD 300 or Master applications.

Table 1-1. Related Documentation

Category	Title	Description
Hardware/Station-Level	Advant Station 5*nn* Hardware User's Guide	There are several Advant Station Hardware user's guides, one for each version of the Advant Station. The *Advant Station Hardware User's Guide* covers the installation, and maintenance of the hardware for an Advant Station.
	Advant Station 500 Series IMS User's Guide & Advant Station Operator Station User's Guide	The *Advant Station 500 Series IMS User's Guide* and *Advant Station Operator Station User's Guide* provide instructions for system administration functions.
AdvaInform Software	AdvaInform Basic Functions User's Guide	The *AdvaInform Basic Functions User's Guide* describes the basic functions that support all AdvaInform software packages, including backup and restore.
	AdvaInform Object Types Reference Manual	The *AdvaInform Object Type Reference manual* provides table names and attribute names. This information is useful when you specify the data to be stored by History or develop AdvaInform User API applications that deal with history.
	AdvaInform Reports User's Guide	The *AdvaInform Reports User's Guide* describes how to include historical information in process reports.
	AdvaInform SPC User's Guide	The *AdvaInform SPC User's Guide* (MOD 300 only) provides instructions for the SPC option that applies Statistical Process Control analysis techniques on the historical data.
	AdvaInform PDL User's Guide	The *AdvaInform PDL User's Guide* (MOD 300 only) provides instructions for the PDL (Production Data Log) option for History.
	AdvaInform Display User's Guide	The AdvaInform Display User's guide describes how to incorporate historical data into displays that you build and run with AdvaInform Display software.
	AdvaInform User API User's Guide	This book describes the procedures and reference information for AdvaInform User API.

Table 1-1. Related Documentation

Category	Title	Description
AdvaInform Software	AdvaInform SQL*Connect User's Guide	This book describes the Client and Server parts of AdvaInform SQL*Connect and how to use the available tools to configure those parts.
	AdvaCommand User's Guide	The *AdvaCommand Basic Functions User's Guide* describes how to use the trend graphs to view History data during runtime.
	AdvaBuild Environment Builder User's Guide	The *AdvaBuild Environment Builder User's Guide* (MOD 300 only) describes how to specify the variables for standard trend displays. It also describes how to configure External Display Access to support History windows on the Advant Station with AdvaCommand software.
	AdvaBuild Display Builder User's Guide	The *AdvaBuild Display Builder User's Guide* describes how to put trend graphs on custom graphic displays.

1.6 Release History

This book describes version 2.2 of AdvaInform History. The new functionality introduced with each release is described below.

1.6.1 Release 1.0

This is the initial release of AdvaInform History. The functionality supported by this initial release includes collection, storage, and retrieval of numeric, message, and report logs.

1.6.2 Release 1.1

Data archival is the new functionality supported by this release. In addition, release 1.1/1 supports Production Data Logs (PDLs).

1.6.3 Release 1.2

This release supports the following new functionality:

- Calculations for primary logs. In prior releases, calculations could only be done on data before storing it in a secondary log.
- A log that uses deadband can be a data source for another log.
- In prior releases, log time period could only be configured on a log group basis. As before, once a log group is configured and saved, the log period can not be changed for the log group. However, in this release, the log time period can be changed on an individual basis for a log (via the Composite Log window).
- Logs can use both calculation & deadband. In prior releases they were mutually exclusive.

- Seamless retrieval makes it easier for applications to access historical data:
  - For trend data presentation, if you scroll back in time beyond the capacity of the specified log, the seamless retrieval function will go to the next (secondary) log in the log hierarchy (if a suitable secondary log exists).
  - For User API, you do not need to know the name of the log in order to retrieve data. Rather you can specify such attributes as log period, calculation algorithm, and retrieval type, and History will search for the log that most closely fits the profile.
- The functionality of the sample blocking rate has changed. The blocking rate is the amount of time that samples are buffered in collection before being calculated (if applicable) and stored in History. Blocking rate is only used in primary logs.

1.6.4 History 2.1

This is a major revision with emphasis on improving History throughput and capacity. The enhancements and new functionality provided in this release are summarized below:

- History data can now be saved in files. This improves performance and increases capacity:
  - Performance increases by an order of magnitude (20,000 entries/minute stored).
  - Capacity increases by using 4 or 24 bytes to store a single entry. This increases historical capacity by a factor of 4 minimum.
- Numeric logs and composite logs are no longer registered as OMF objects. This lets you create configurations of up to 100,000 logs.
- A monthly storage interval is available.
- dbLoad has been improved to report disk capacity required for the given configuration.
- A new utility (hsBAR) allows off-line backup and recovery of all historical data.
- Timed archive of reports is supported.
- Start state of archive groups is configurable. Defaults to START_ACTIVE. Other choices are START_INACTIVE and START_LAST_STATE.
- A new collection type (USER_SUPPLIED) is provided that allows synchronous logs to accept data from user inputs rather than collect it.
- New logs can be added to existing composite logs.
- Dual History is a means of distributing a composite log on two nodes. These logs on each node run in parallel with each other.
- dbDump is able to read a list of tags from a file and create a data base load file.

1.6.5 History 2.1/1 (MOD 300 Only)

This is an incremental release of AdvaInform History. The enhancements and new functionality provided in this release are summarized below:

- The MOD 300 Message log and Message Form are redesigned.
- Up to 250,000 entries are allowed per log.

1.6.6 History 2.2

The enhancements and new functionality provided in this release are summarized below:

- History Status Display

  The Runtime Status Display shows the status of all configured logs. Thus you can determine from one display if any logs are inactive. Prior to this release, you were required to look at each log group or log set display to get this information.

  The History Log Status window is an overview page. It shows the total number of logs on the node, the number of active logs, inactive logs and logs in pending state, and it will control access to the Detailed Log Information display.

  The Detailed Log Information display shows the log name, log status, and some key collection and storage attributes for each log.

  The log status is updated automatically. Also, when logs are created or deleted, the Runtime Status display indicates that these changes have occurred.

  You can activate and deactivate logs from the Runtime Status display.

- hsBAR back up with only one tape and to a remote tape drive.
- ability to display log data from AdvaInform History on a Multibus-based console in systems with MOD 300 software. Refer to the Advant Station 500 Series IMS User's Guide or Advant Station Operator Station User's Guide for details on how to configure the Advant OCS data base to support this functionality.
- Recalculate limits on SPC logs.

1.7 Terminology

AdvaInform Basic Objects

AdvaInform Basic Objects are pre-defined object types for storage of AdvaInform data. They can be stored by History and can be used in AdvaInform applications. They include:

DI, DO	Bit, 0/1, On/Off, Boolean objects
AI, AO	Floating Point objects
TXT	String objects
DAT	Boolean, Integer (Word (2 bytes), or Intlong (4 bytes)), or Floating Point objects
DATA_FCM	Floating Point objects (MOD 300 only)
VA_STRING_FCM	String objects (MOD 300 only)

You need the AdvaInform Object Handling package to use Basic Objects This package contains a configurator to instantiate the basic Objects. It also executes the objects during runtime.

Advant OCS objects

See **Object** below.

AdvaInform User API

User Application Programming Interface.AdvaInform User API in AdvaInform Basic Functions allows programs in the HP-UX environment to interact with Advant OCS objects.

API (Application Programming Interface)

A set of subroutines that provide a means to establish remote process-to-process communication.

Attribute

The public data of an object. The result of the application and the configured data is commonly stored in an attribute. Other objects can request the object's public data through the Core System using subscriptions. AdvaInform User API applications can request attribute values via either the Object Services or the SQL Services.

Controller

This refers to the Advant Controller which handles I/O and executes control logic in the Advant OCS.

Control Network

A network connecting different types of Advant stations/controllers used to supervise and control a specific part of the plant. It is possible to exchange data between a Control Network and the Plant Network and also between different Control Networks via the Plant Network (Plant Network, see below).

Command

Operation requests to an object. As a result, the object performs the requested action. AdvaInform User API can give commands to objects via object services.

Core System

All nodes in the distributed control system communicate with each other through the Core System. The Core System provides services for the system applications. Using these services, the system applications can communicate with each other whether they reside in the same node or in different nodes of different types. The Type Directory, which is part of the Core Directory, contains information on all object types known to the Advant OCS.

DCS

Distributed Control System

ES

Engineering Station

External Node

An external node is a node that is not part of the Advant OCS. Advant Stations and Controllers are internal nodes (part of the Advant OCS). An external node can access AdvaInform Basic Functions via the X-protocol or via SQL*Net.

IMS

Information Management Station

IMS Menu

The menu that provides access to AdvaInform Basic Functions as well as optional functions.

Objects

In AdvaInform software (as well as other Advant technology software) the combination of data and associated procedures (commands and/or operations that can be applied to the data) are represented by **Objects**. Objects represent significant elements or functions in the process control/process automation domain. Applications are created by combining these objects. Examples of objects are:

- Process objects such as CCF loops (MOD 300) or PIDCON and AI (Master)
- History logs
- Alarm and event lists
- AdvaInform Basic Objects
- AdvaInform User Objects

Objects protect some of their data for the object's internal use only. Other object data, called *attributes*, are public and available to other objects or software applications. For example, attributes of CCF PID or PIDCON loop objects include the measured value, setpoint value, and base gain.

The procedures that tell an object what to do are called operations. There are two types, normal operations and commands. Normal operations specify the algorithm for the object. For example, the normal operation of a PID loop is the PID algorithm. A command is an operation that involves an object or User API program making an operation request of another. For example, a supervisory object can command a PID loop to change its setpoint to a new value.

Objects can read attributes in other objects via subscriptions. These subscriptions can read data from the source objects either cyclically, on demand, or when a predefined event occurs.

User Objects

User objects are a means to develop and integrate user applications in AdvaInform. These objects are built using the optional AdvaBuild Object Type Builder software. They are instantiated and executed using the optional AdvaInform Object Handling software.

Process Objects

Process Objects such as PID Controller FCMs and Analog Input FCMs (MOD 300) or AI, AO, and PIDCON (Master) represent control and supervision functions that may exist in Controllers.

History Object

History objects support historical execution. Examples of history objects are logs and log managers. History objects are built with the History Builder.

Log

A log is a history object. It is the basic storage unit in the historical database. A log can contain a series of values extracted from a particular data source. This data is collected based upon the configuration of attributes, which define the log's function.

1.8 Product Overview

AdvaInform History supports collection, storage, retrieval, and archival of process data such as process variable values and alarm/event messages. This historical data is available for use in other applications. For instance:

- It can be displayed on AdvaCommand trend displays.
- It can be displayed on Multibus-based trend displays (MOD 300 only). To configure the Advant OCS database to support this functionality, refer to the *Advant Station 500 Series IMS User's Guide* or *Advant Station Operator Station User's Guide*.
- It can be analyzed by AdvaInform SPC (MOD 300 only)
- It can be included in reports produced by AdvaInform Reports.
- It is available for custom software applications via SQL (Structured Query Language) and User API (Application Programming Interface).
- It can be presented on AdvaInform Display client nodes.

1.8.1 Historical Logs

Historical data is stored in *logs*. The basic structure of a log is illustrated in Figure 1-2.

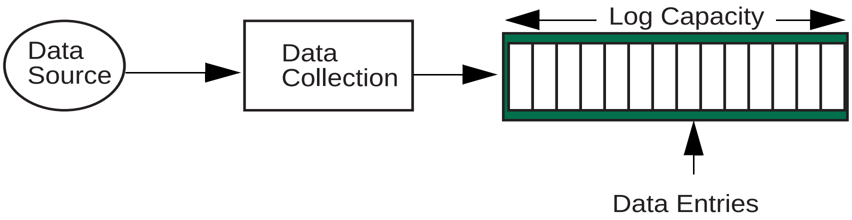


Figure 1-2. Historical Log

Each data source requires its own log. A log is located entirely in one node. Data sources for the log can be located on the local node or other nodes on the Control Network.

There can be thousands of logs in a historical application. Different log types are provided for different types of data. For instance, process values are stored in numeric logs, messages are stored in message logs, and so on.

A log holds multiple data values called *entr*ies. An entry is identified by a time stamp. When the log is full, the oldest entries are replaced by new entries.

For example, Log_FC101 is configured to hold data values from flow loop FC101. Since the storage interval is 10 seconds, every tenth second a new entry is added to the log. Each entry has a time stamp to identify the time (including date) that the sample was taken, Figure 1-3.

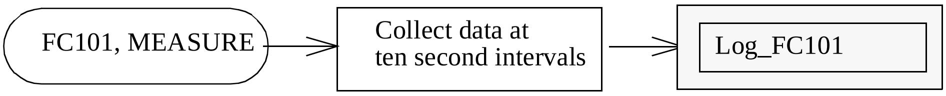


Figure 1-3. Example, Data Storage for Flow Data

Logs are the basis of the History Software. All aspects of History involve either configuring, collecting, and storing data in logs, retrieving data from logs, or archiving/restoring logs.

1.8.1.1 Log Hierarchy - Composite Logs

History logs can be connected to form log hierarchies, Figure 1-4 (MOD 300 example) and Figure 1-5 (Master - TTD example). All logs are contained within a **composite log**. Composite logs can have primary and secondary logs. A *primary log* gets its data directly from a data source. A secondary log gets its data from either a primary log or another secondary log.

For example, in Figure 1-4, primary log $HSTC100, MEASURE-1-o samples data from the measured variable of loop TC100. In turn, $HSTC100,MEASURE-1-o serves as the data source for secondary log $HSTC100,MEASURE-2-o which serves as the data source for $HSTC100,MEASURE-3-o. The storage rate of a secondary log is a multiple of the storage rate of its source log.

The composite log structure provides the following benefits:

- Data retrieval is easy via the access name (data source). By default, all logs in a composite log take on the data source name as their access name. This allows data to be retrieved by the data source name instead of the log name.
- Secondary logs can be used to store other aspects of the logged variable such as maximum and minimum values for a specified time interval.
- Different views of a process variable are possible via logs with different time spans (for instance, high resolution data for a short time span in a primary log, and a lower resolution for a longer time span in a secondary log).
- You can store data over longer periods without using as much storage capacity.

Other composite log configurations are shown in Figure 1-6 and Figure 1-7.

NOTE

All logs in a composite log must reside on the same node. It is not allowed to have a secondary log on a different node than the primary log. The only case where two logs in a composite log can reside on different nodes is the dual history configuration, Figure 1-7.

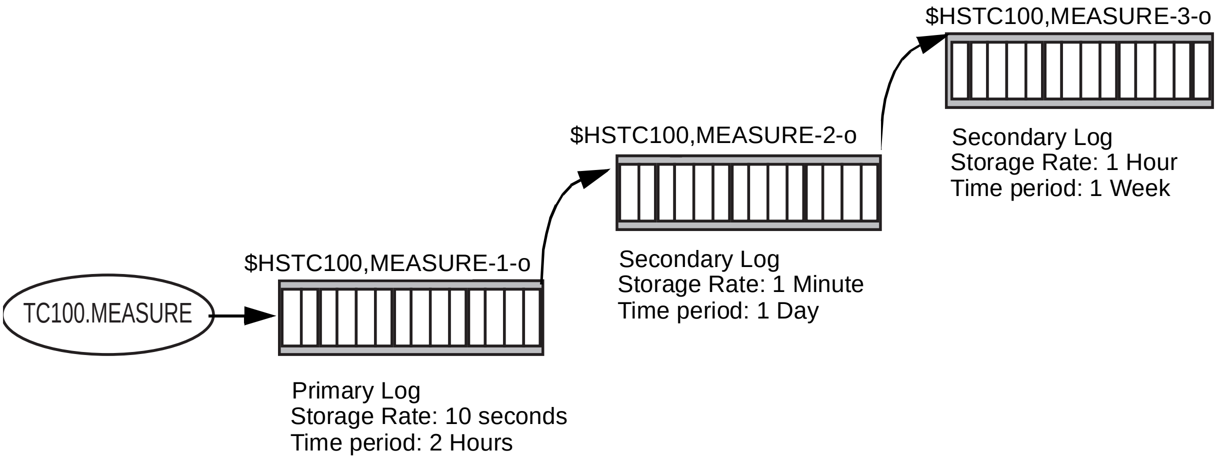


Figure 1-4. Example Composite Log, MOD 300

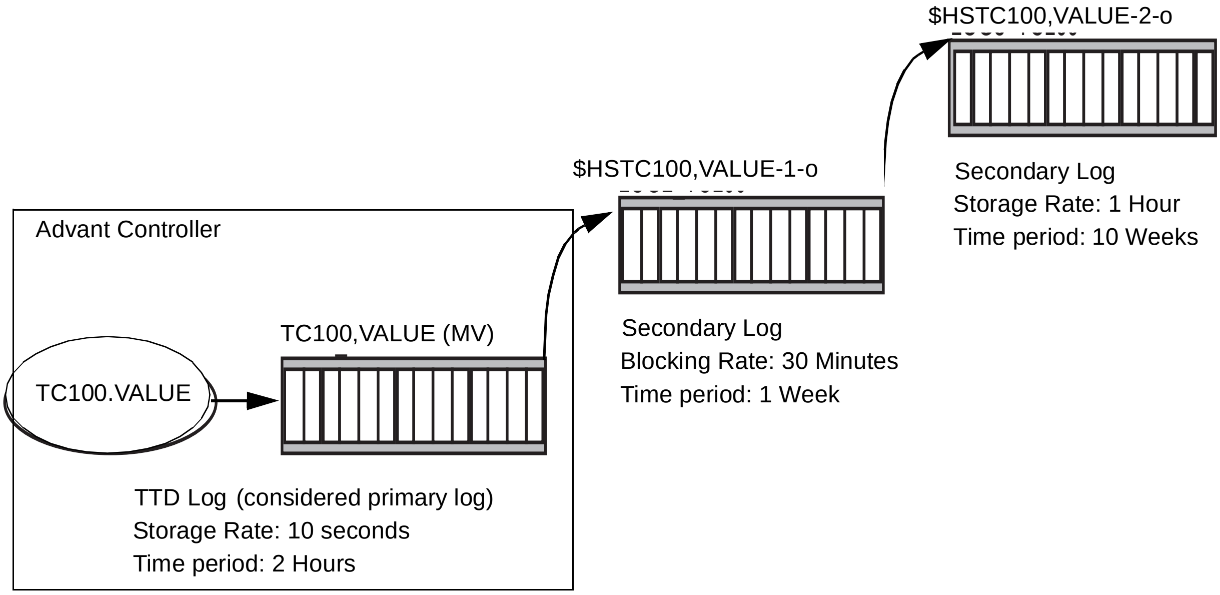


Figure 1-5. Composite Log, TTD Example

Primary Log as Data Source for Multiple Secondary Logs

A log can be the data source for more than one secondary log, Figure 1-6.

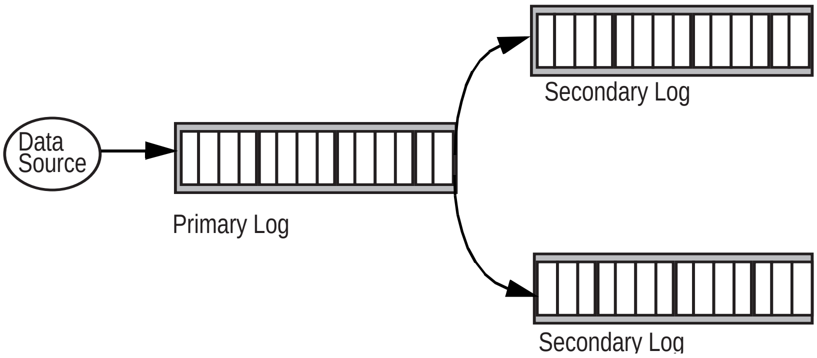


Figure 1-6. Secondary Logs with the Same Data Source

Dual History - MOD 300 Only

The dual history configuration shown in Figure 1-7 is only applicable for MOD 300. Dual history refers to the case where a composite log has two primary logs that can reside on different nodes. Both primary logs must receive data from the same data source as shown in Figure 1-7.

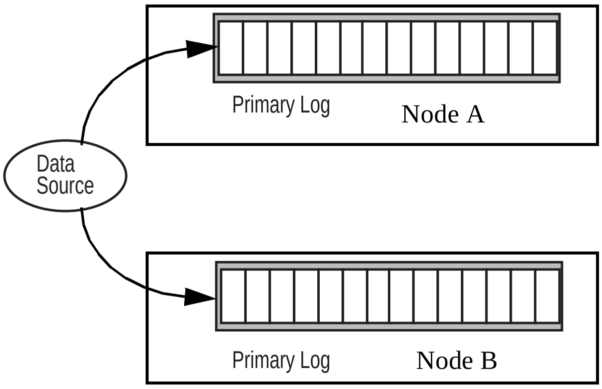


Figure 1-7. Composite Log with Two Primary Logs

This is the only case where logs in a composite log can reside in different nodes. Both logs are accessible by access name. History will determine where the data resides.

1.8.1.2 Log Groups

The log group is used to define a group of logs with common storage parameters. Every log must belong to a log group. A log group specifies the following characteristics for a log:

- history node where the log resides
- log type such as numeric or message
- basic data collection attributes including storage interval, collection mode, and log period
- As an option, a log group can reference a log typical which is a template for many similar logs (attribute values are configured the same).

All logs in a log group reside on the same node, they are all of the same log type, and they all have the same collection mode, storage interval, and period. These attributes can not be changed once they are defined. The one exception is log period which can be changed on an individual basis for logs within a group.

Data collection can be started (activated) or stopped (deactivated) for all logs in the group by one command issued to the group by the operator or by a User API program.

1.8.1.3 Log Sets

The log set structure lets you start or stop data collection for a number of logs simultaneously with one command. Log groups also provide this functionality; however, with log sets you have more flexibility in the logs that you assign to it. The only restriction is that all logs in a log set must reside in the same node. Thus logs in the same log set can have different collection attributes (storage interval, and so on).

You are not required to assign a log to a log set. Do so only if you require the functionality of the log set. A log can only belong to one log set; however, it may be assigned to both a log group and a log set.

1.8.1.4 Log Typicals

A log typical is a re-usable template for building many similar logs (logs that collect Historical data for similar loops). You can assign a log typical to a log group such that all the logs in that group will have the same default attribute values. Then, if required, you can edit the attributes for logs on an individual basis. If you modify a log typical, the changes you make to that log typical do not affect existing logs that you created using that log typical.

1.8.1.5 Composite Log Typicals

Composite log typicals are re-usable templates for building two or more similar log hierarchies. To build a composite log typical, you first make a log group for each component log of the hierarchy and then you connect the groups together via the Composite Log Typical window.

1.8.1.6 Log Types

The following log types are supported.

Numeric log	Values of process variables. Values may be integer or floating point; however, they are always retrieved as a floating point number.
SPC	Calculated SPC subgroup statistics derived from entries of a numeric log. (MOD 300 only)
PDL	Information collected from a batch execution (MOD 300 only). Entries can be either numeric values, messages, or associations to other logs and resources.
Report log	Holds the output of an AdvaInform Report. Entries in a report log can be: Output from a finished report. This provides the structure to modify a finished report and save both the original unchanged and the modified report. It also provides storage for viewing the report later, and it provides the ability to archive the report. Output can be character-based, PostScript, HTML, or PDF. Encapsulated data. This is data produced by a report execution for inclusion in another report. This type can also be archived.
Message log	Alarm and event messages.

1.8.1.7 Log Attributes

Each log has attributes that define its operating characteristics such as collection criteria, log size, and so on. All log types have some common attributes, as well as attributes that are unique to that log type. As an example, Table 1-2 shows a partial listing of attributes for a numeric log (approximately 25 additional attributes are not listed). A description of the interaction of the attributes and complete lists of attributes are provided in Section 3.1, Design Considerations.-

Table 1-2. Partial Listing of Numeric Log Attributes

Name	Description	Valid Entries/States
Log Name	The log name which must be unique system wide.	A text string, 80 characters maximum.
Description	Log description.	A text string, 40 characters maximum.
Access Name	The name (other than the log name) for accessing this log. May be associated with the process object name.	A text string, 65 characters maximum.
Log Group	Name of the Log Group containing this Log.	Name of a Log Group
Log Set	Name of the Log Set containing this log.	Name of a Log Set
Archive Group	Name of the Archive Group containing this Log.	Name of an Archive Group

When you configure History, you build a hierarchy of logs, and specify the log attributes. For a numeric log, you specify its object name, description, and other attributes via a History Builder. The operation of a log is often determined by the interaction of multiple attributes. For example, data collection for a numeric log depends upon the values of its Sample Interval, Storage Interval, Blocking Rate, Compaction, and Calculation Algorithm.

1.8.1.8 Log Entries

A log entry holds several kinds of information about the log, depending on how the log is configured. For instance, a log entry typically consists of a Time Stamp, Data Value, and Status, Figure 1-8. What information is included and what information is not included in a log entry is described in Section 3.1.4, File Storage vs. Oracle Tables.

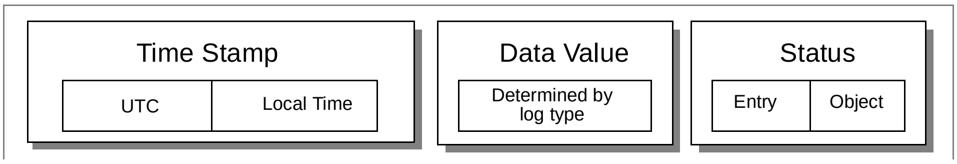


Figure 1-8. Log Entry Format

1.8.2 History Database Configuration

History database configuration is done in part via the History Builder, and in part via the dbDump/dbLoad functions. You use the History Builder to configure such History objects as Log Groups, Log Sets, Log Typicals, Report Logs, Message Logs. You also use the History Builder to configure one template log for each basic numeric log that fits a certain pattern (for example one log with a 10-second storage interval and one with a 6-second storage interval). The dbDump and dbLoad functions are then used to create the rest of your numeric logs based on those templates.

1.8.3 Data Collection

History collects samples from various data sources. Data sources can be process objects such as AI and PIDCON attributes for Master, CCF loops or analog input FCMs for MOD 300, AdvaInform basic objects, user objects, other historical objects, as well as data in other databases via SQL*Connect server functionality. They can be in any node of the system. A typical historical application stores the values of thousands of individual data sources.

A common application of history is to periodically store values of process variables such as flows, temperatures, or tank levels. For example, to store the value of a steam flow, your data source is the measured variable of the loop that controls the flow.

Besides the analog variables mentioned above, history can store the following types of data:

- Digital values
- Alarm and event messages
- Operator actions such as parameter changes
- Messages generated by Taylor Control Language (TCL) such as billboard messages or notification of program status (MOD 300)
- Event and alarm messages generated by process objects in controllers, or by Basic or User objects in the Advant Station (Master)
- User entered data such as lab data.

- Data entered through the AdvaInform User API (Application Programming Interface).
- Batch data accumulated during a batch production run.
- Finished reports produced by AdvaInform Reports software

Data collection attributes are configured via the History Builder. For guidelines on how to configure these attributes, refer to Section 3.1.3, Data Collection Considerations and Section 3.1.3.8, Data Collection Examples.

1.8.4 Data Storage

1.8.4.1 File Storage vs. Oracle Tables

Log entries can be stored in either Oracle tables or directly in files maintained by History. The benefit of file storage is that it is faster and uses less disk space than Oracle storage. File-based storage is only available for synchronous numeric logs. Asynchronous numeric logs, message logs, report logs, SPC logs, and Production Data Logs (PDLs) are always stored in Oracle tables.

How to configure synchronous numeric logs for either file-based or Oracle table storage, and related considerations are described in Section 3.1.4, File Storage vs. Oracle Tables.

NOTE

It is strongly recommended that you use file-based storage when possible.

1.8.4.2 Log Capacity

How History manages log capacity depends on the type of log:

Numeric Logs and Message Logs	The log is created to hold a certain number of entries. When that limit is reached, the oldest values are overwritten by newer values. This is referred to as wrap-around.
Report Logs	Once the capacity is reached, the oldest report is deleted each time a new one is added.

1.8.5 Data Retrieval

Data retrieval refers to the presentation of historical data on operator displays and access of historical data by other applications such as User API, AdvaInform SQL*Connect, and AdvaInform Reports.

1.8.5.1 Data Presentation

Data Presentation is supported by AdvaCommand software trend displays. These displays are simple graphs of process variables versus time, Figure 1-9. Data presentation is also supported by AdvaInform Display, and Multibus-based trend displays.

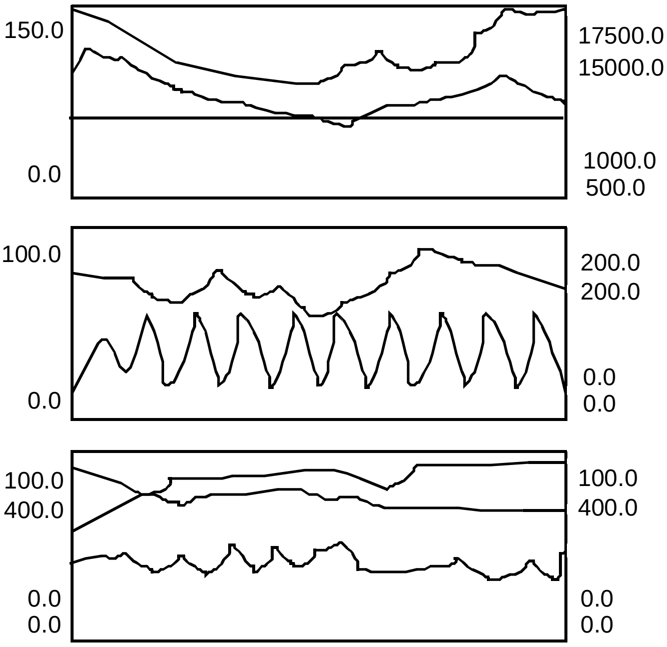


Figure 1-9. Historical Data on a Trend Display

The AdvaCommand software contains standard trend displays. All you have to do to configure these displays is to specify the variables you wish to see on them. In MOD 300, you do this via the Group and Area definition phases of the AdvaCommand Environment Builder. AdvaCommand also has a trend element which allows you to put trend graphs on displays you define with the AdvaBuild® Display Builder.

All logs in a composite log are accessible by the composite log name (access name). History provides a seamless retrieval feature where the History software determines which log in a multiple-log hierarchy to retrieve data from, based on time and other criteria.

In MOD 300, the data on the trend display may come exclusively from the CCF trend buffer, exclusively from History logs, or a combination of CCF trend buffer and History logs. The following rules determine which data is used for trend display:

- When a request for data is made from the group trend and CCF trending is configured, all points come from CCF.
- If CCF trending is not configured, plotted points are from History and there will be a gap at the end of the trend that is equal in size to the blocking rate.
- When a request is made for data at the current time, History will combine log and trend buffer data. The data from the trend buffer will fill in the gap between the latest entry stored in History and the current time.

The trend display provides the means to scroll forward and backward in time. If you scroll backward in time and go beyond the limit of the history log that is currently being accessed, the seamless retrieval function will automatically switch to the next log that contains older data in the log hierarchy (if your log hierarchy is configured to have a log with older data).

History can return the raw data that is actually stored, or interpolated data where entries are calculated at fixed intervals. This is useful when you use deadband and not all the entries are stored. INTERPOLATED will ensure the expected number of values are returned.

1.8.5.2 Other Applications that use Historical Data

AdvaInform SQL*Connect Access of History Logs

History logs are objects in the Advant OCS database. As with all Advant OCS objects, they can be read by queries from SQL (Structured Query Language). This functionality is supported by the *AdvaInform SQL*Connect* software which makes Advant OCS objects appear as relational data base tables to SQL. This feature is used by the AdvaInform Reports package when it includes History data in reports. It is also used when History data is read by vendor applications from host computers attached to the History node via the plant network. Information on how to read numeric logs is provided in Section 3.1.6, Data Retrieval Considerations.

AdvaInform User API Access of History Logs

History logs can also be accessed via AdvaInform User API applications. You can use AdvaInform User API to develop applications with 3-GL languages such as C. These programs contain functions which read from or write to, or otherwise manipulate Advant OCS objects. User API is described in the *AdvaInform User API User's Guide*.

AdvaInform Reports

History supports AdvaInform Reports by providing historical information for the reports. For example, reports can access numeric, message, and PDL data.

Production Data Logs (MOD 300)

Production Data Logs (PDLs) are an option for AdvaInform History software. PDLs support collection, storage, and retrieval of production data in Batch 300 and TCL applications. PDLs are described in detail in the *AdvaInform Production Data Log User's Guide*.

Unlike other log types in AdvaInform History, you do not configure PDLs. When the PDL option is installed on an Advant Station with AdvaInform software, PDLs are created automatically by the application (Batch 300 or TCL).

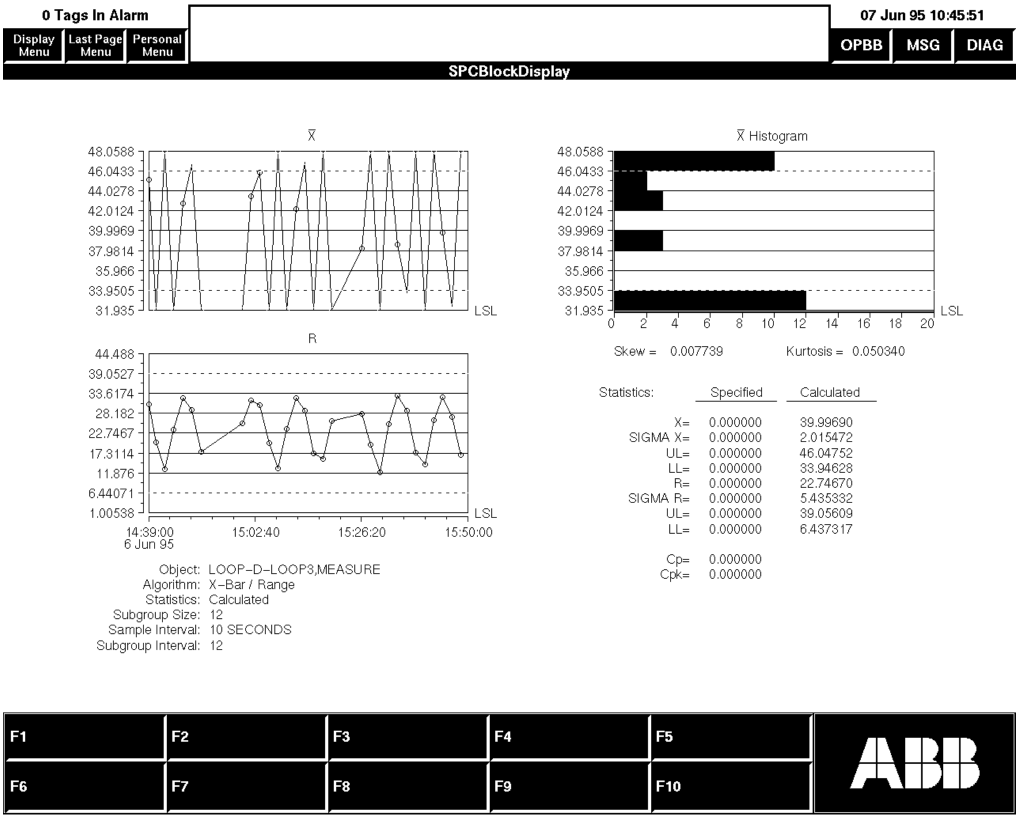
Data storage for PDLs is controlled by TCL statements and the Recording section of the Site Recipe Editor in Batch 300. You can access data stored in PDLs via SQL queries as described in the *AdvaInform Basic Functions User's Guide*. In addition you can build custom reports or use pre-defined PDL reports to access PDL data. General instructions for building and executing reports are provided in the *AdvaInform Reports User's Guide*. For information regarding batch reports that use PDL data, refer to the *Batch 300 User's Guide*.

PDLs can be archived as a completed report via the AdvaInform Reports archival function, or raw PDL data can be archived via the History archival function.

You can display a listing of PDLs for a given history node, and delete PDLs via the History Builder. PDLs can also be deleted automatically.

AdvaInform SPC (MOD 300)

Numeric logs provide the inputs for AdvaInform SPC software. SPC stands for statistical process control which is an analysis technique that indicates if a process is producing consistent results. History stores raw numeric values from process variables in a log. The SPC algorithms are applied to this data and the results are stored in another log. These values are available for display as SPC charts such as the one shown in Figure 1-10. Refer to the *SPC User’s Guide* for details regarding SPC.



Skew = 0.007739

Kurtosis = 0.050340

Statistics:	Specified	Calculated
X=	0.000000	39.99690
SIGMA X=	0.000000	2.015472
UL=	0.000000	46.04752
LL=	0.000000	33.94628
R=	0.000000	22.74670
SIGMA R=	0.000000	5.435332
UL=	0.000000	39.05609
LL=	0.000000	6.437317
Cp=	0.000000	
Cpk=	0.000000	

Object: LOOP–D–LOOP3,MEASURE
Algorithm: X–Bar / Range
Statistics: Calculated
Subgroup Size: 12
Sample Interval: 10 SECONDS
Subgroup Interval: 12

Figure 1-10. SPC Chart Using Historical Data

1.8.6 Data Archival and Restore

When the log is full, the oldest entries are replaced by new entries, or are deleted. You can maintain a permanent record of all entries for a particular log or set of logs through archival. When you archive, the log entries for a specific log (or set of logs) for a specific time period are copied from the history database to the DAT tape.

Related logs can be assigned to an archive group such that all the logs in the archive group are archived according to the same archive parameters. Before you can use the archive functions you must configure the archive device (DAT tape drive). If you plan to use archive groups you must configure the archive groups and assign logs to them. The following functions are provided for handling archive data.

Initialize Archive Media

You must initialize the media before you can store archive data on it. To do this you select an archive device (DAT drive), and then assign a Volume ID and Volume Label to identify the tape.

Archive Logs

You can archive (copy) logs from the run time history database to an archive media (DAT). The run time history database is where on-going (current) storage of history data occurs. When you archive logs, the log data is copied from the run time portion of the history database to the archive media. In addition, an archive entry is recorded in the archive database. The archive entry indicates the archive time, number of logs in the entry, archive group, and other data.

Restore Logs

You can restore archived logs from the archive media to the history database. When you restore archive logs, they are stored in the restored portion of the history database.

Delete Restored Logs

You can delete restored logs from the restored history database.

Add Archive Entries to the Database

When you restore logs from the DAT drive to the History database, it is helpful to have the corresponding archive entry information for those logs (Volume ID, Label, start and stop times and so on). To facilitate restoration of archived logs, you can add archive entry records from the archive database on the archive media to the archive database on the History node internal drive.

View the Archive Entries

You can view archive entries stored in the archive database on the disk. You can also view archive entries on the archive media currently mounted on the archive device.

Delete Archive Entries

You can delete archive entries from the archive database on the History node internal drive.

1.8.7 History Status

The History Status function provides a convenient method for monitoring numeric log status, and activating/deactivating numeric logs on a log list basis. Access to history status information and functions is via by the History Log Status main window, Figure 1-11.

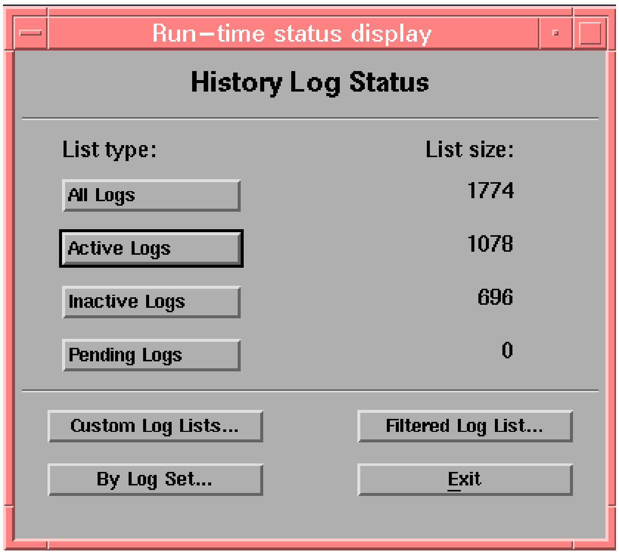


Figure 1-11. History Log Status, Main Window

This main window provides access to detailed log status windows for viewing History log status for:

- all logs
- all active logs
- all inactive logs
- pending logs
- log sets
- custom log lists

You can activate and de-activate logs from these detailed log status windows. You can also build custom log lists from the detailed log status windows.

1.9 User Interface

The user interface for AdvaInform History is based on HP Common Desktop Environment (HP CDE). For information regarding this user interface, refer to the *HP Common Desktop Environment User's Guide*.

Chapter 2 Installation

2.1 Site Planning Environment

Refer to the *Advant Station 5nn Hardware User's Guide*.

2.1.1 Guidelines for planning installation of Advant Station

Refer to the *Advant Station 5nn Hardware User's Guide*.

2.2 Setup

Generally, AdvaInform History software is loaded onto the Advant Station at the factory prior to delivery. Therefore, you are generally not responsible for loading the software. If you are required to load the software for some reason (for instance, if you purchase AdvaInform History software after initial delivery of the Advant Station), refer to the *Advant Station 500 Series IMS User's Guide* (or *Advant Station Operator Station User's Guide)* for instructions.

CAUTION

Make sure that your system is properly set up for clock synchronization and daylight saving, and the time is set correctly before you start any History logs. An Advant OCS workstation should be the master timekeeper. Changes to the system time after logs have been activated will result in loss of data.

2.2.1 External Display Access (MOD 300)

When you use AdvaInform History on an Advant Station with AdvaCommand software, External Display Access must be configured to support History displays in the AdvaCommand environment via the Environment Builder. Refer to the *AdvaBuild Environment Builder User's Guide* for details.

2.3 Shut-down Procedures

The AdvaInform History run time environment is automatically stopped when the Advant Station is stopped. This procedure is covered in the *Advant Station 500 Series IMS User's Guide* and *Advant Station Operator Station User's Guide*.

2.4 Start-up Procedures

The AdvaInform History run time environment is automatically started as part of the Advant Station start-up. This procedure is covered in the *Advant Station 500 Series IMS User's Guide* and *Advant Station Operator Station User's Guide*.

When you use AdvaInform History on an Advant Station with AdvaInform software, the History window is activated from the IMS menu. How to open this window is described in Section 3.3, Application Start-up.

2.5 Product Verification

Open the History window as described in Section 3.3, Application Start-up. Then open the History Configuration window as described in Section 3.6.3, History Configuration Window. From this window, choose **Help - On Version** from the **Help** menu. Verify that the version corresponds with the delivery information. Also verify that the local History Manager is listed as active. Choose **Exit** from the **File** menu to close the window.

Chapter 3 Configuration and Application Building

The History Builder is a flexible configuration tool with many features. You can design simple data collection applications such as recording a single process variable, or you can build complex applications using composite logs with calculations and data compaction. This chapter is organized to help you configure the most basic History applications without getting bogged down in complicated functions that you do not require. It also provides guidance for complex History applications. Briefly, this chapter is arranged in the following manner:

- Section 3.1, Design Considerations

  This section describes important characteristics of logs that you should understand before you begin your configuration. It provides a listing of all the attributes that you can define for logs to determine how the logs operate, and it describes how to define key attributes to achieve such functions as data compaction, and so on. You may want to refer back to this section when you actually configure the logs.

- Section 3.3, Application Start-up

  This section describes how to access the History Builder.

- Section 3.4, Tutorial

  This section provides some quick and straightforward procedures for configuring basic History functions. The tutorials, listed below, are intended for you to get some hands-on experience, learn the basics of the History Builder without getting bogged down in details.

  - Section 3.4.2, Tutorial 1 - Storing Numeric Values at a Periodic Rate

    This is the most basic History application (no composite logs, typicals, calculations, or data compaction).

  - Section 3.4.3, Tutorial 2 - Building Log Hierarchies Using a Composite Log Typical

    This application demonstrates how to implement some more advanced historical functions such as log typicals and composite logs.

- Section 3.5, Application Procedures

  This section provides guidelines for the following application procedures:

  - Section 3.5.2, Create History Directories
  - Section 3.5.3, Log Sets
  - Section 3.5.4, A Simple Log without using Typicals
  - Section 3.5.5, Simple Logs using a Log Typical
  - Section 3.5.6, Composite Logs using a Log Typical and Composite Log Typical
  - Section 3.5.7, Configure History Node to Support Archive Function
  - Section 3.5.8, How to Build Report Logs
  - Section 3.5.9, How to Build Message Logs for MOD 300
  - Section 3.5.10, Configure Logs to be Archived with Log Typicals

  - Section 3.5.11, How to Edit Log Attributes
  - Section 3.5.12, Error Messages
  - Section 3.5.13, How to Configure History Logs in a Text File
- Section 3.6, Configuration/Application Menus

  This section provides a detailed description of the function and operation of each window, dialog, and menu item related to History configuration.

3.1 Design Considerations

3.1.1 Recommended Method for History Database Configuration

It is recommended that you use the dbDump and dbLoad functions to configure the bulk of the numeric logs in your History database. This method is far more efficient than configuring the entire database using the History Builder. Even when you use dbDump and dbLoad, you must use the History Builder to configure such history objects as log groups, log sets, log typicals, report logs, and message logs. In addition you must use the history builder to configure one representative log for each numeric log that fits a certain pattern. These representative logs are then used as templates to create the rest of the numeric logs via dbDump/dbLoad.

3.1.2 Data Sources for Numeric Logs

In MOD 300, the most common data source for a numeric log is a CCF loop. There are two formats for identifying loop attributes as data sources. If the attribute is at the loop level, as defined by CCF, the format is: tag,attribute where tag is the name of the loop and attribute is the CCF attribute name. For example,

TC500,MEASURE

If the attribute is on the FCM level, the format is: tag.fcm, attribute where fcm is the name assigned to the FCM. For example

LP500.MTH,RESULT

Other objects whose data type is either real or integer can be used as a data source for a numeric log. The format (normal case for Master) is: object,attribute where object is the object name and attribute is an attribute from that object. For example

AI_1,VALUE

In systems with Master software, digital input (DI) objects can be used as data sources. Compaction can be applied to DI logs. TTDs can also be data sources. The following specification can be used to get just the state (ON/OFF) bit of the DI status:

DI_1,VALUE

3.1.3 Data Collection Considerations

The data collection function is illustrated in Figure 3-1. This function collects values and buffers them internally for a period of time according to the sample blocking rate. It then performs the specified calculation on the buffered data. A value is calculated for every storage interval. These calculated values are passed to the deadband function which checks to see if the value is within or outside the deadband range. If deadband is not active, the value is sent to storage. If deadband is active, the value is sent to storage if any one of the following are true:

- The value is outside the deadband range.
- The value is within the deadband range, but the deadband storage interval time has elapsed since the last value was stored.
- The value status changes (possible statuses are: no data, bad data, good data)

If none of these criteria are met, the value is not stored.

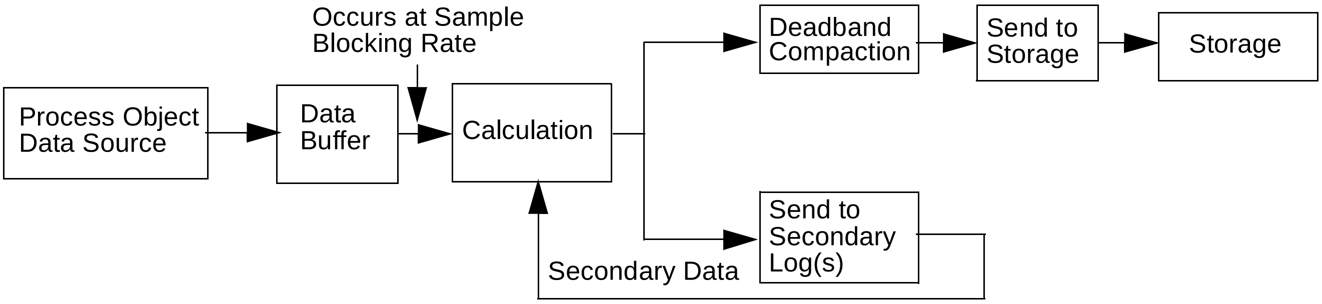


Figure 3-1. Data Collection, Functional Diagram

Figure 3-1 shows that the data sent to a secondary log does not undergo data compaction, even if the primary log has deadband configured. The secondary log gets all values that result from the calculation. Thus, the calculation performed by the secondary log is more accurate than if done using just data stored after deadband compaction.

3.1.3.1 Collection Mode and Collection Type

The collection mode determines how data is processed when it is received by History. The collection mode can either be synchronous or asynchronous:

- Synchronous - Time Sequence Data. It is often used for process variables and is usually selected for numeric logs. The time interval between samples received by History must always be the same. Calculations and deadband can be applied to synchronously collected data. There are two collection types of synchronous logs:
  - API: History periodically *collects* data from a data source. Secondary logs must be configured as API synchronous.
  - USER_SUPPLIED: Samples are *sent* to History from a user application such as a User API program. The time between samples sent is expected to be equal to the sample rate of the log.

  There is no difference in functionality between API and USER_SUPPLIED.

- Asynchronous - data is entered by an application such as AdvaInform User API or TCL. This mode is only for primary logs. Message logs and report logs must use Asynchronous mode. The time between entries sent to History does not have to be the same every time.

You can configure History to perform calculations on collected data before the data is actually stored in the log. When a calculation is active, the specified algorithm, such as summing or averaging, is applied to a configured number of data samples from the data source. The result of the calculation is stored by History. For example, you can configure History to collect five values from the data source, calculate the average of the five values, and then store the average. This calculation is actually configured by time, not the number of values. The calculation is applied to the values collected over a configured time period.

You can also configure History for data compaction which uses a mathematical compaction algorithm to reduce the amount of data stored for numeric variables. Both calculations and deadband can be used concurrently for a log. The calculation is done first. Then the deadband algorithm is applied to see if the result should be stored.

If the time stamp of a received value is greater than or equal to 50% before, or greater than or equal to 100% after the sample interval, the status of the stored data is NO_DATA. Time stamps may drift due to load in the system. An example of a case where time drift can cause NO_DATA entries to be stored is illustrated in Figure 3-2.

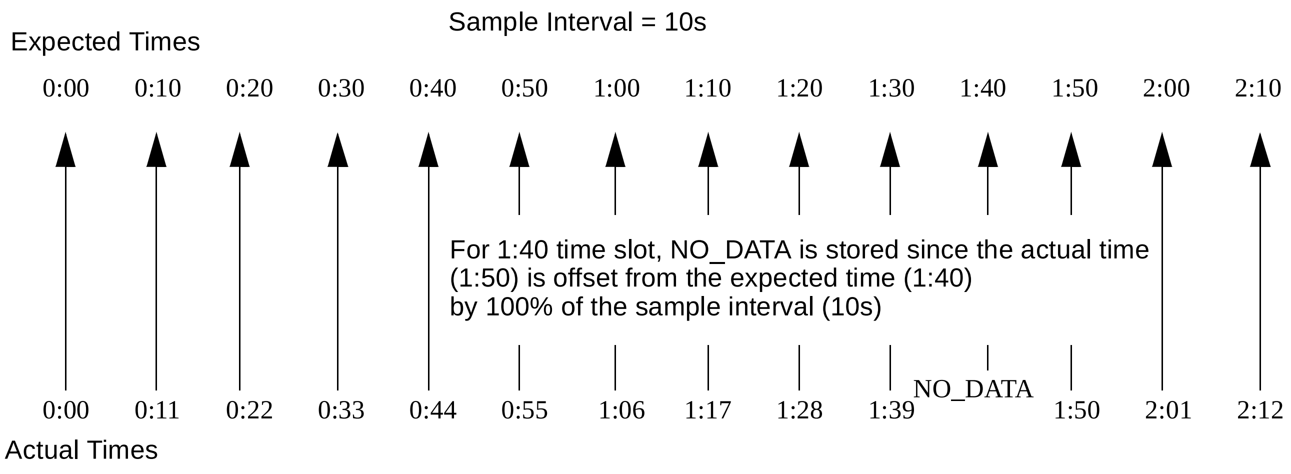


Figure 3-2. Example of Time Drift

The attributes that determine how data is collected in the synchronous Collection Mode are:

- Collection Type
- Sample Interval
- Calculation Algorithm and Storage Interval
- Blocking Rate
- Activation
- Log Period
- Compaction Deadband and Deadband Storage Interval

Collection Type (described in Section 3.1.3.1, Collection Mode and Collection Type) determines whether History collects the data, or if it is sent to the log and processed as an asynchronous log. All of the following characteristics are applicable to USER_SUPPLIED as well as API collection types.

3.1.3.2 Sample Interval

The sample interval is defined on an individual basis for each log. A data sample is taken at each sample interval. For USER_SUPPLIED, this is the expected time interval between samples sent to History. If a sample is not collected at the expected time, a NO_DATA sample is created as shown in Figure 3-2. Whether or not the sample is stored depends on whether or not a calculation algorithm is specified, and whether or not a compaction deadband is specified.

3.1.3.3 Calculation Algorithm and Storage Interval

You can specify calculations to be done on data before it is stored in a log (either primary or secondary), Figure 3-3. This allows you to store a single value that is representative of a larger time span of values, or key characteristics of the data. For example, you can sample a variable every minute and put those values into a primary log. Then you can configure that an hourly average, minimum, and maximum be calculated from data in the primary log and store those values in secondary logs.

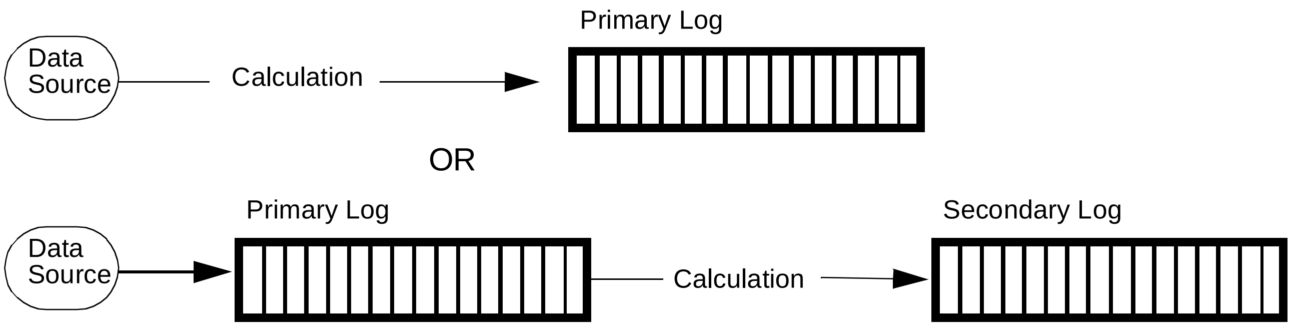


Figure 3-3. Calculations

The type of algorithm is specified via the Calculation Algorithm attribute. The choices are: Instantaneous (no calculation), Average, Maximum, Minimum, Sum, Sum of squares, Standard deviation, Number of values.

The storage interval is defined on a log group basis. If a calculation algorithm other than INSTANTANEOUS is specified for the log, the algorithm is performed at each storage interval. The storage interval must be a multiple of the sample interval. For example, if the sample interval is 5s (five seconds), the storage interval can be 10s, 15s, 20s, 25s, and so on.

When the calculation is performed, the specified number of data samples are entered in the algorithm, and the result is passed to the deadband function. If the calculation algorithm is INSTANTANEOUS (no calculation), the sample and storage intervals must be equal, and each data sample is treated as a result.

3.1.3.4 Blocking Rate

Blocking rate is only used in primary logs. It is used in different ways, depending upon the type of data source. Most synchronous primary logs use blocking rate to allocate a buffer in the collection function to hold the blocking rate worth of samples before applying a calculation and (if applicable) deadband algorithms and sending the samples to storage. Each blocked sample consumes 28 bytes of memory.

The exception to this is TTD logs (Master). A major function of the Blocking Rate when TTD is the data source is to limit the communications load placed on the RTA board in the History node and the controller (source of the process data being logged). For TTD logs, collection is not required to allocate any additional memory for blocking rates. TTD logs buffer data in the node where they exist. When a blocking rate is configured for a tag with a TTD source, the collection function uses the blocking rate to trigger a collection from the TTD at intervals equal to the blocking rate. For example, if the sample interval of the log is 1 minute and the blocking rate is 10 minutes, collection will request 10 samples from the TTD log every 10 minutes.

The blocking rate can be used to phase data from logs with similar sample intervals by using prime number multiples of the sample interval. For example, consider a system that has 1000 logs. If all 1000 logs have a 10 minute blocking rate, every 10 minutes storage would be flooded with 1000 messages. This could have a negative affect on performance. Instead you can use blocking rate to distribute the load over two or more intervals. For instance you can assign 500 logs a 9-minute blocking rate, and the other 500 logs an 11-minute blocking rate. This way, the flood of 1000 messages would be limited to once every 99 minutes (9x11), rather than once every 10 minutes.

3.1.3.5 Activation

You can also use the first Activation time to distribute the load. Both the blocking rate and activation time may be defined initially when you configure the History database. If you do not configure these attributes initially, or if you find that adjustments are required, the Stagger Collection function in the hsDBMaint utility provides a convenient means for adjusting these log attributes on-the-fly. Refer to Section 5.6.14, Stagger Collection and Storage.

It is recommended NOT to use small differences (less than one minute) to distribute the load. Managing log collection at slightly different times will result in significant overhead.

3.1.3.6 Log Period

Disk space is allocated to logs according to their log capacity, which is a function of the log time period and storage interval. The log period can be adjusted on an individual log basis to keep log capacity at a reasonable size for each log. For all synchronous logs with or without data compaction, log capacity is calculated as: log capacity = (log period/storage interval)

3.1.3.7 Compaction Deadband & Deadband Storage Interval

Compaction has two advantages:

- It reduces the load on the system by filtering out values that do not have to be stored since there is not a significant change in the value.
- It allows data to be stored on-line for a longer period of time since the data is not replaced (overwritten) as quickly.

Compaction is specified on an individual log basis. You may use it for some logs and not for others. Compaction should not be used indiscriminately in all cases.

- If you want an accurate representation of your process over time, and you do not require that every data sample be stored, you can use compaction.
- If your application requires that every data sample be stored as an instance in the log, you cannot use compaction.

Deadband can be used on digital data sources so that only state changes are saved.

The Deadband filters out insignificant changes in the process variable.The deadband defines the minimum change in value or in rate of change for the change to be significant. The deadband is defined as a percent of the engineering units range.

For example, if a process variable is scaled from 200 to 500, the range is 300. If the deadband is 2%, the minimum significant change is 6 (2% of 300). With these values, if the last stored sample was 400, and the next sample is 397, then 397 is within the deadband and will not be stored. If the next sample is 406.2, it will be stored. A small deadband results in more stored samples than a large deadband.

Deadband is activated by configuring a compaction deadband greater than zero. When deadband is active, the deadband algorithm determines whether the calculation result (or raw value if calculation algorithm is instantaneous) is within or outside the deadband. If the most recent result is outside the deadband, the previous result is stored. If the most recent result is within the deadband, it is held until the next result is calculated, and the previous result is discarded. If the compaction deadband is zero (compaction not activated), all results are stored.

A deadband storage interval is defined on an individual basis for each log. The deadband storage interval is a multiple of the log's storage interval. It is the maximum time that can elapse without storing a sample when compaction is activated. Even if the process is so stable that there are no samples outside the deadband, samples are still be stored at the deadband storage interval. The deadband storage interval is always measured from the time the last sample was stored. The deadband storage interval should be at least 10 times larger than the storage interval to ensure a high compaction ratio.

When deadband is active, a sample may be stored for any one of the following reasons:

- sample outside the deadband
- value status changed (possible statuses are: no data, bad data, good data)
- deadband storage interval elapsed since last stored sample

Figure 3-4 shows how samples are stored when deadband is active. The deadband function holds each sample until it determines whether or not the sample is required to plot the trend. Samples that are not required are discarded.

In Figure 3-4 data compaction is activated at 0s, so a sample is stored. At 10s, the sample is held until the next sample is taken at 20s. Since there is no change in slope between 0s and 20s, the 10s sample is discarded. At 30s a change in slope is detected, so the sample at 20s is stored to plot the change. Again, the 30s sample is held until the next sample is taken at 40s. Since there is no change in slope between 30s and 40s, the 30s sample is discarded. At 50s another change in slope is detected, so the sample at 40s is stored.

Even though there are spikes between 40s and 60s, these values were not sampled so the slope remains constant (as indicated by the dotted line in Figure 3-4). Another change in slope is detected at 70s. The sample at 50s is discarded, and the sample at 60s is stored. The slope remains constant thereafter. The next sample stored is 100s since the Deadband Storage Interval (40s) elapses between 60s and 100s.

When you use compaction, the log period is effectively increased so that the same number of samples stored (as determined by the log capacity) cover a longer period of time. Consider the following two cases. If your process is very unstable such that every sample is stored and there is no data compaction, the effective log period will be equal to the configured log period.

If your process is perfectly stable (maximum data compaction, samples stored only at deadband storage interval), the effective log period is extended according to the following formula:

effective log period = (deadband storage interval/storage interval) * configured log period

In most cases, since the amount of process variation will not be at either extreme, the effective log period will fall somewhere in between the configured (minimum) and maximum value.

The calculated effective log period is stored in the EST_LOG_TIME_PER attribute. This attribute is updated periodically according to a schedule which repeats as often as every five days. If necessary, you can manually execute the update function via the hsDBMaint utility. This is described in Section 5.6.5, Update Deadband Ratio On-line.

When requesting by access name, EST_LOG_TIME_PER is used to determine which log to retrieve data from (seamless retrieval). For instance, consider a composite log where the primary log stores data at 30 second intervals over a one-day time period, and the secondary log stores data at one-hour intervals over a one-week time period. Your data compaction ratio is 5:1 which extends the effective log period of the primary log to five days. If you make a request to retrieve data that is three days old, the History retrieval function will know to get the data from the primary log rather than the secondary log, even through the configured log period of the primary log is only one day.

When EST_LOG_TIME_PER is updated, both internal History memory and the history database are updated with the new values of EST_LOG_TIME_PER and COMPACTION_RATIO. This will ensure the information will be available after a re-start of the History node.

Since the spikes at 42s and 48s do not occur on a Sample Interval, they are not detected, nor are they reflected in a trend display. The trend from 40s to 60s would be displayed as indicated by the dotted line. A smaller storage interval would be required to provide a more accurate representation of the trend between 40s and 60s.

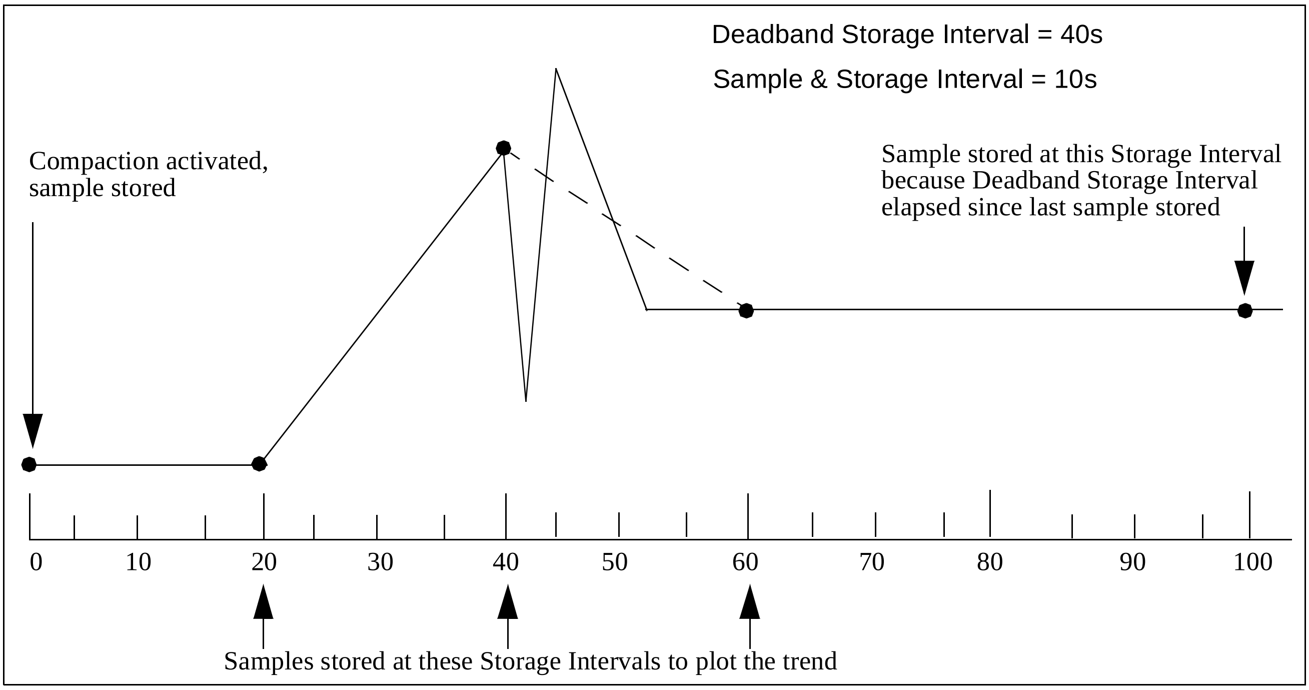


Figure 3-4. Data Compaction Example

Before configuring compaction, refer to examples 2 & 5 in Section 3.1.3.8, Data Collection Examples.

3.1.3.8 Data Collection Examples

These examples are valid both for Master and for MOD 300. For Master, the general recommendation is to use TTD logs as input and to be very restrictive with direct logging from Advant Controller objects at least with low sampling intervals. The reason is TTD logging gives much higher performance.

Example 1, Storing Instantaneous values, no compaction

When the calculation algorithm is INSTANTANEOUS (no calculation), the sample and storage interval must be the same. If no compaction is specified, a data instance is stored every storage interval. Figure 3-5 shows the relation between these attributes of the Primary Log. In this case a sample is stored every five seconds. The blocking rate is 60 seconds. This means that the data samples to be stored are held in a buffer, and then written to disk in blocks of 12 samples at each 60-second interval. While the data is held in the buffer, it is not available for data retrieval.

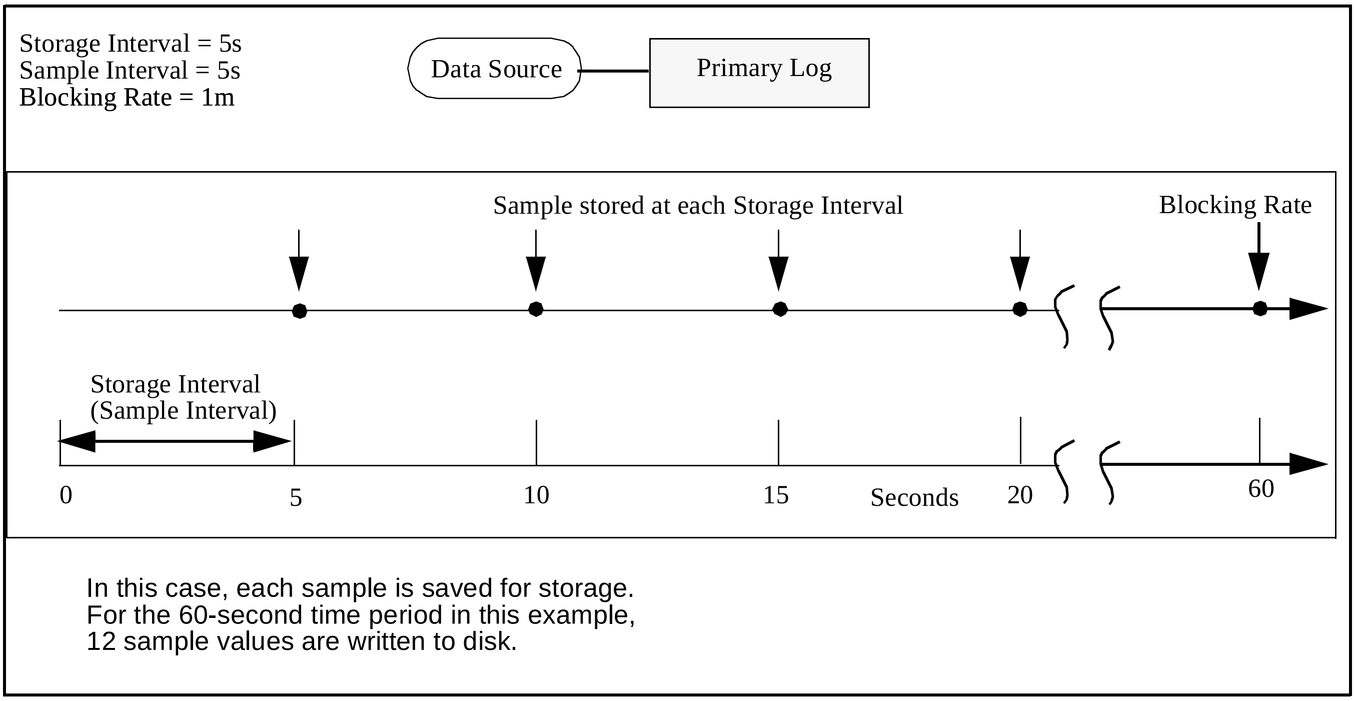

Storage Interval = 5s
Sample Interval = 5s
Blocking Rate = 1m
Data Source
Primary Log
Sample stored at each Storage Interval
Blocking Rate
Storage Interval
(Sample Interval)
0
5
10
15
Seconds
20
60
In this case, each sample is saved for storage.
For the 60-second time period in this example,
12 sample values are written to disk.

Figure 3-5. Storing Instantaneous Values, No Compaction

Example 2 - Storing instantaneous values using compaction

In this example, the process variable is scaled from 200 to 500, the range is 300. The deadband is 2%, the minimum significant change is 6 (2% of 300). Again, since there is no calculation, the sample and storage interval must be the same. At each storage interval the data is checked by the deadband function. If a significant change in slope is detected, the sample is stored. Samples are discarded as the slope remains within the deadband. However, if the deadband storage interval (in this example, 25s) elapses since the last stored sample, another sample is stored on the deadband storage interval whether or not a change in slope is detected.

In Figure 3-6, the data is sampled every five seconds. Of the 17 samples taken, only 7 are stored. A summary is provided in Table 3-1 below.

Table 3-1. Storing Instantaneous Values w/Compaction

Interval	Value	Stored	Reason
0	406.7	Yes	Deadband activated, first sample stored
5	404.5	No	No significant change in slope between 5s & 10s
10	406.2	Yes	Change in slope, next sample outside deadband
15	399.8.1	No	No significant change in slope between 15s & 20s
20	394.9	Yes	Change in slope from 20 to 25, sample @ 25 outside deadband
25	409.0	Yes	Change in slope from 25 to 30, sample @ 30 outside deadband
30	404.2	Yes	Change in slope from 30 to 35, sample @ 35 outside deadband
35	404.0	No	No significant change in slope between 35s & 40s
40	402.8	No	No significant change in slope between 40s & 45s
45	403.3	No	No significant change in slope between 45s & 50s
50	403.2	No	No significant change in slope between 50s & 55s
55	404.1	Yes	Deadband storage interval (25s) elapsed since last stored sample (at 30s)
60	404.1	No	No significant change in slope between 55s & 60s
65	403.8	No	No significant change in slope between 60s & 65s
70	404.1	No	No significant change in slope between 605 & 70s
75	402.9	No	No significant change in slope between 70s & 75s
80	405.0	Yes	Deadband storage interval elapsed since last stored sample

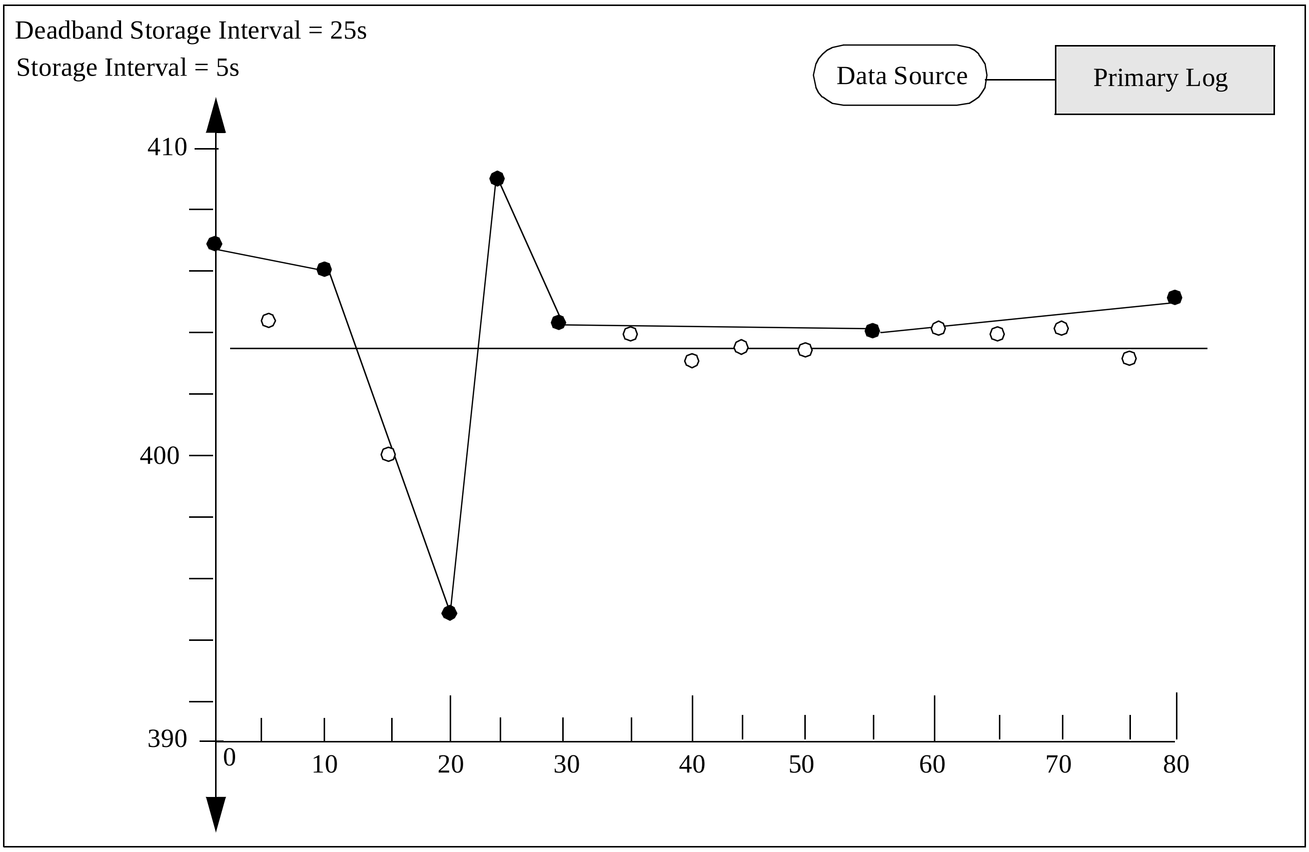


Figure 3-6. Storing Instantaneous Values w/Data Compaction

Example 3 - Storing Calculated Values

This example shows how to do a calculation before storing a value in a primary log, Figure 3-7. In this example, a shift (eight-hour) average is calculated.

The storage and sample intervals determine how many samples to use in the calculation (storage interval/sample interval). The storage interval must be an integer multiple of the sample interval. In this case, eight samples are used in each calculation.

Eight samples at hours 1,2,3,4,5,6,7, and 8 are averaged and the average is stored in the primary log at hour 8 (storage interval). Similarly, eight samples for two other sample groups are averaged and stored. The results of these three average calculations are written to disk at 24 hours (blocking rate). The process is repeated at 48 hours and averages are stored at 32, 40, and 48 hour.

.

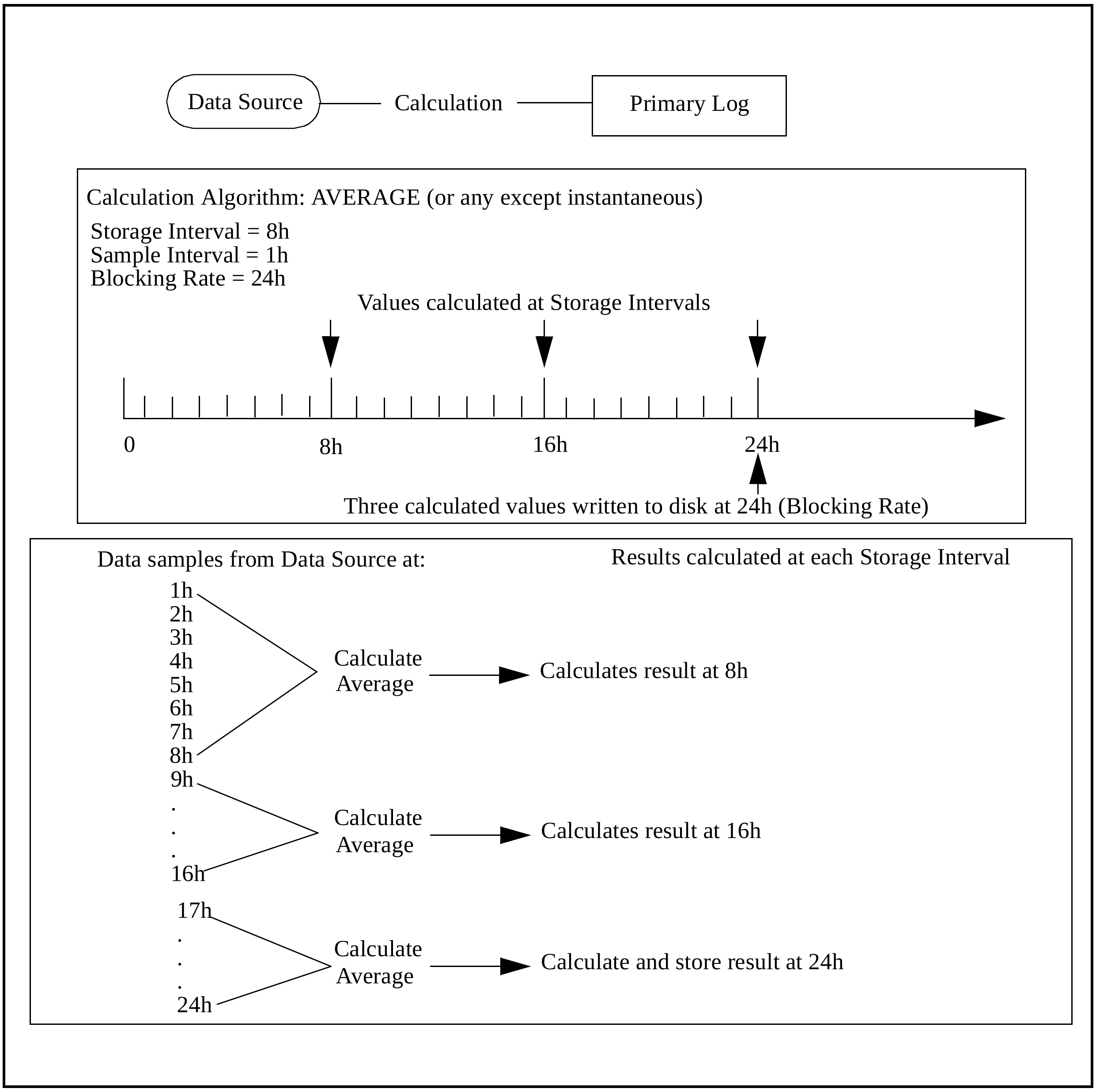

Data Source
Calculation
Primary Log
Calculation Algorithm: AVERAGE (or any except instantaneous)
Storage Interval = 8h
Sample Interval = 1h
Blocking Rate = 24h
Values calculated at Storage Intervals
0
8h
16h
24h
Three calculated values written to disk at 24h (Blocking Rate)
Data samples from Data Source at:
Results calculated at each Storage Interval
1h
2h
3h
4h
5h
6h
7h
8h
Calculate Average
Calculates result at 8h
9h
.
.
.
16h
Calculate Average
Calculates result at 16h
17h
.
.
.
24h
Calculate Average
Calculate and store result at 24h

Figure 3-7. Primary Log Example: Calculation being Used

Example 4 - Storing Calculated Values in Primary and Secondary Logs

This example shows how to do a calculation before storing a value in a secondary log. The primary log calculates the shift average as described in Example 3, and shown in Figure 3-7. The secondary log then calculates a daily average as illustrated in Figure 3-8.

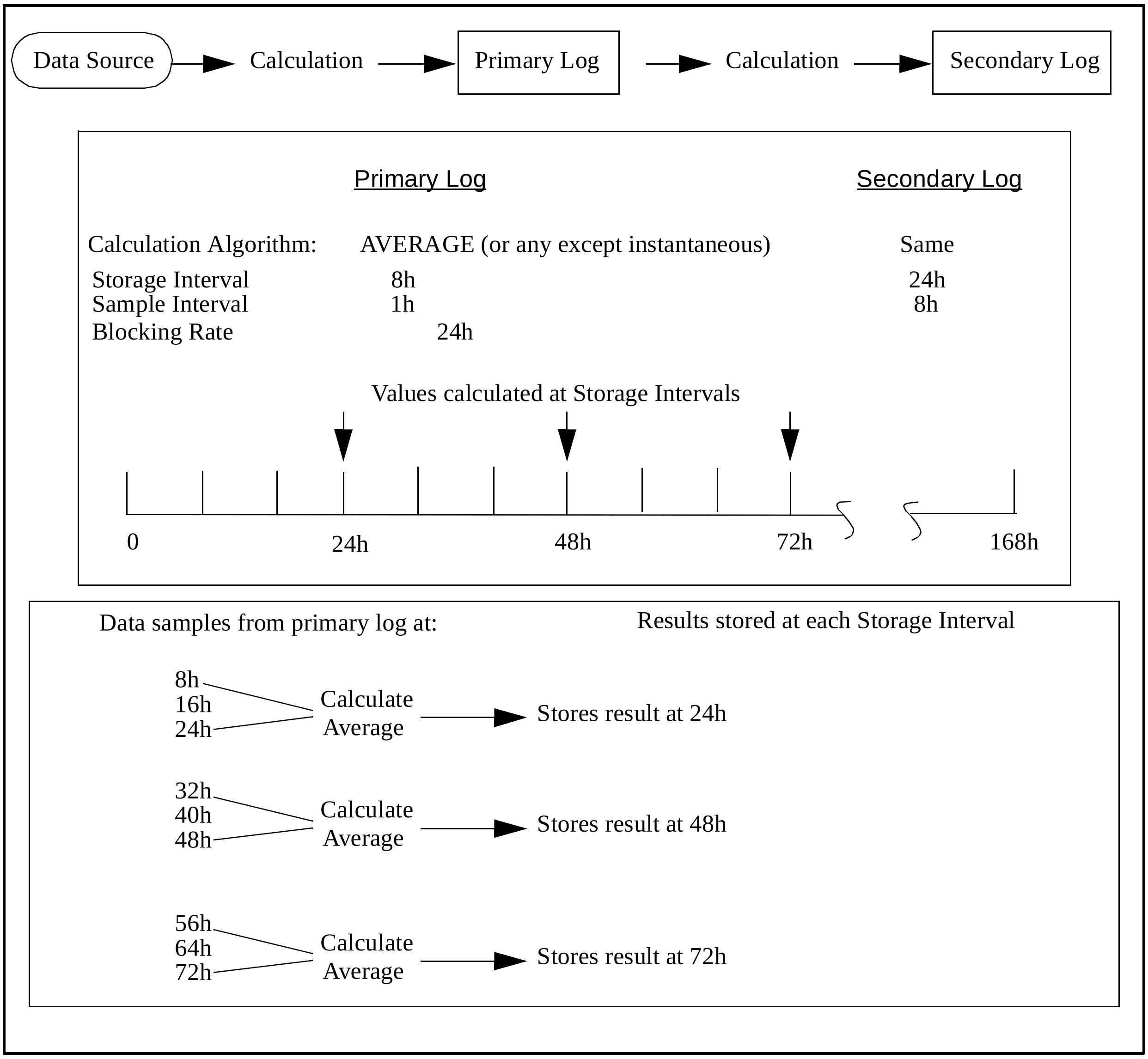


Figure 3-8. Secondary Log Example: Calculation being Used

NOTE

The sample blocking rate is not used for secondary logs. The sample blocking rate from the primary log determines when values are stored for all logs in the composite log.

Example 5- Storing Calculated Values with Compaction

This example uses the same data as Example 2 - Storing Instantaneous Values using Compaction. The process variable is scaled from 200 to 500, the range is 300. The deadband is 2%, the minimum significant change is 6 (2% of 300). Since a calculation is being performed at each storage interval, the storage interval must be an integer multiple of the sample interval. In this case, data samples are collected at 5-second intervals, and four samples are used to calculate an average at 20-second intervals. At each storage interval the calculated value is checked by the deadband function. If a significant change in slope is detected, the calculated value is stored. Calculated values are discarded as the slope remains within the deadband. However, if the deadband storage interval (in this example, 60s) elapses since the last stored sample, another sample is stored at that time whether or not a change in slope is detected.

In Figure 3-9, calculations are performed at 20-second intervals. Of the 4 calculations performed, only 2 are stored. A summary is provided in Table 3-2 below.

Table 3-2. Storing Calculated values w/Compaction

Interval	Value	Calculated Average	Stored	Reason
5	406.7	404.3	Yes	First calculated average
10	404.5			
15	406.2			
20	399.8.1			
25	394.9	403.0	No	No significant change in slope between 1st & 2nd calculation
30	409.0			
35	404.2			
40	404.0			
45	402.8	403.4	No	No significant change in slope between 2nd & 3rd calculation
50	403.3			
55	403.2			
60	404.1			
65	404.1	403.7	Yes	Deadband storage interval (60s in this example) elapsed since last stored sample (0s)
70	403.8			
75	404.1			
80	402.9			

Notice the contrast in the trends between Figure 3-6 and Figure 3-9, even though the same data is used to plot the trends.

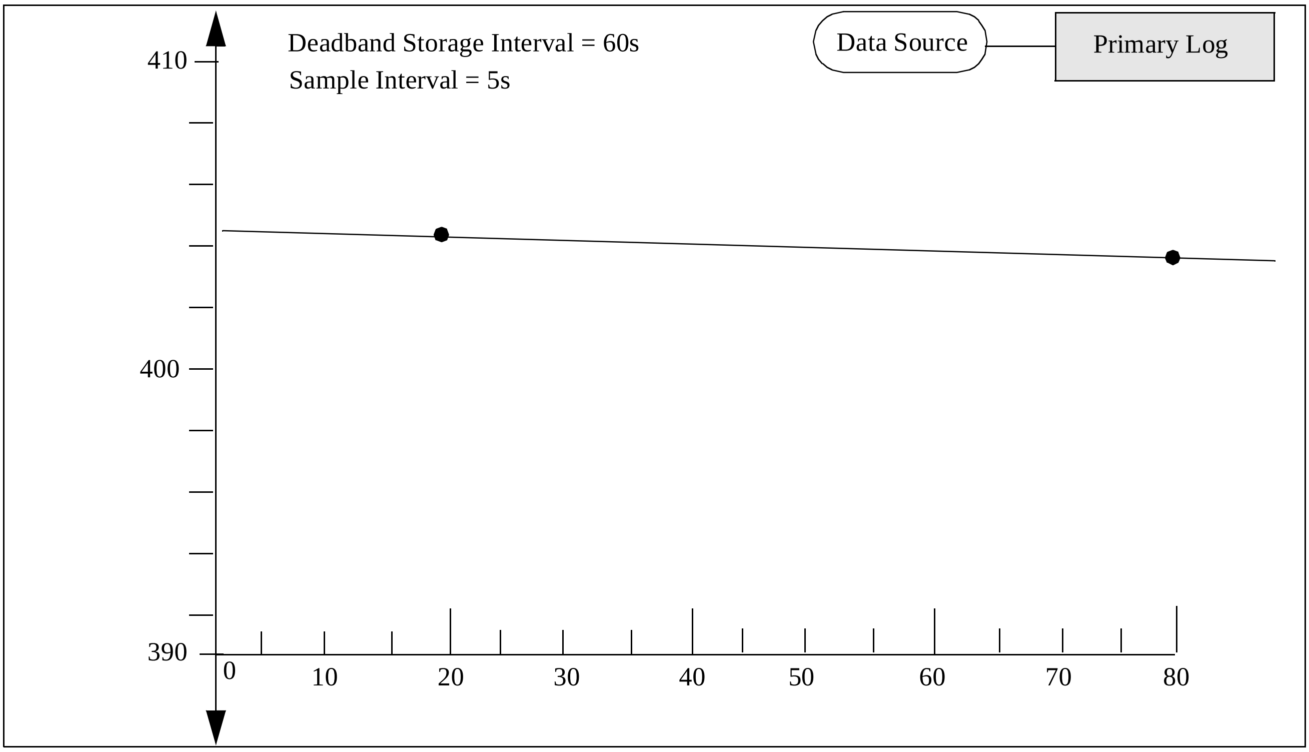


Figure 3-9. Storing Calculated Values w/Data Compaction

3.1.4 File Storage vs. Oracle Tables

Log entries can be stored in either Oracle tables or directly in files maintained by History. The benefit of file storage is that it is faster and uses less disk space than Oracle storage. Whether you store entries in Oracle tables or in files maintained by History is determined by the STORAGE_TYPE log attribute:

- ORACLE = storage in Oracle tables @ 100 bytes per entry
- TYPE1 = file storage @ 24 bytes per entry
- TYPE2 = file storage @ 4 bytes per entry

For details regarding disk space requirements, see Section 3.1.4.2, Disk Requirements for Oracle-based Logs and Section 3.1.4.3, Disk Requirements for File-based Numeric Logs.

3.1.4.1 Comparison of TYPE1 and TYPE2 File-based Logs

The default storage type is TYPE1 (24-byte). The 24-byte entry provides the full range of status and related information for each entry. When you select the TYPE2 storage type, each entry requires less storage space; however, you sacrifice some of the status information provided for TYPE1 entries as well as some accuracy with regard to time stamps. Also, for TYPE2 entries, if a value is modified, the original value is not saved, and no status bits are set to indicate that the value has been modified.

Use of the TYPE2 storage type should be limited to TTD logs. For example, TTD does not have an object status, and it also sets time stamps to the storage interval. Therefore when collecting from TTD, TYPE2 should be used since very little functionality is lost.TYPE2 storage type may be appropriate for logs that have fast sample rates (one minute or faster), if you determine that the degree of inaccuracy with regard to time stamp is small enough to be acceptable.

The full range of information provided for TYPE1 entries is described below. The differences between TYPE1 and TYPE2 entries for each category of information are described where applicable.

Time Stamp

Each data entry in a log is time stamped. This stamp consists of two different times:

- Universal Time (UTC) Coordinate time stamp is Greenwich Mean Time. It is not affected by the time zone nor daylight savings time. This gives consistency throughout all external changes such as Daylight Savings Time and any system resets that affect the Local time.
- Local Time is the actual time of day dependent upon time zone and daylight savings time.

For TYPE2 logs (file-based 4-byte entries), the time is an estimate of the actual time for the entry. The time for the entries are all defined to be a storage interval apart. So if a log is configured to store a value every 15 seconds, the entries will have timestamps at 0, 15, 30, 45s of each minute.

Data Value

The data value format depends upon the log type. Numeric logs can hold integer and decimal numbers.

Entry Status

The entry status information is a bit set. Each bit provides status information regarding the log entry as described in Table 3-3. Note that while the complete range of information is stored for TYPE1 logs, TYPE2 logs do not store all status bits. The absence of a status value indicates that the data is good.

Table 3-3. Description of Bit Set for Entry Status

Bits in HsLogStatusType	Bit number	Description	TYPE1	TYPE2
noData	1	No data was collected and stored for the entry at this time. Probably due to communications problems between History and the data source.	Yes	Yes
badData	2	Value stored in this entry has bad data. Probably due to a calculation that contains No Data entries in it.	Yes	Yes
entryArchived	3	Not used.	N/A	N/A
valueInAlarm	4	Not used.	N/A	N/A
alarmPriority0	5	Not used.	N/A	N/A
alarmPriority1	6	Not used.	N/A	N/A
timeQuality	7	Not used.	N/A	N/A
timeChange	8	Not used.	N/A	N/A
sourceTimeStamp	9	Time stamp in the entry was obtained with the value collected. Values collected by History usually use the time stamp from the source.	Yes	No
timeStampAdded	10	History generated the time stamp. This is generally done for No Data entries.	Yes	No
_dummy0	11 - 12	Not used.	N/A	N/A
userEntered	13	User entered the data using the AddNumericValueEntry operation.	Yes	Yes
modifiedData	14	When requesting current values from History, this is set if the user modified the original value. If a value has been modified, the No Data and Bad Data bits can not be set. It is assumed the modified value is correct.	Yes	No
modifiedAvailable	15	When requesting original values from History, this is set if the user modified the original value. It indicates that a modified value is available.	Yes	No
_dummy1	16	Not used.	N/A	N/A
firstEntry	17	This bit is set when a log is activated. It does not represent a real value (No Data will also be set). It is a place holder in the log that indicates when the log was activated.	Yes	Yes
lastEntry	18	This bit is set in the status of the last entry collected when a log is deactivated.	Yes	Yes

Table 3-3. Description of Bit Set for Entry Status

Bits in HsLogStatusType	Bit number	Description	TYPE1	TYPE2
valueInterpolated	19	Value returned by History was interpolated during retrieval. It is not a value that is physically stored in the data base.	Yes	Yes
_dummy2	20	Not used.	N/A	N/A
sequenceNum0	21	These bits are used to indicate the sequence number of the log the entry was retrieved from. For example, if the entries were retrieved from log $HSTIC100,VALUE-5-o, then sequenceNum0 and sequenceNum2 would be set.	Yes	No
sequenceNum1	22		Yes	No
sequenceNum2	23		Yes	No
sequenceNum3	24		Yes	No
sequenceNum4	25		Yes	No
TTDValue (Master)	26	This bit is set if the entry was returned from TTD.	Yes	Yes
CCFValue (MOD 300)	27	This bit is set if the entry was returned from CCF Trend Buffer.	Yes	Yes
_dummy3	28 - 32	Not used.	N/A	N/A

Object Status

Object status is the status of the data source when the value was sampled.This field is not used when the data source is TTD. The format of the status (what is represented and how) is dependent upon the data source object type. When a secondary log is created, the object status is obtained from the last primary log entry used in the calculation.

If a User Object that inherits a Basic Object is the source for a log, its status will also be collected and stored.

Object status is not stored for TYPE2 logs.

3.1.4.2 Disk Requirements for Oracle-based Logs

Each numeric Oracle-based log requires approximately 100 bytes per entry:

- 50 bytes for the Inform_HS_RunTime tablespace
- 50 bytes for the HS_Indexes tablespace for logs whose capacity is > 5000 entries, or 20 bytes for the HS_Indexes tablespace for logs whose capacity is < 5000 entries

Each SPC log requires approximately 102 bytes per entry:

- 53 bytes for the Inform_HS_RunTime tablespace
- 49 bytes for the HS_Indexes tablespace for logs whose capacity is > 500 entries, or 20 bytes for the HS_Indexes tablespace for logs whose capacity is < 500 entries

Each message log requires 190 bytes per entry.

After calculating your requirements for Oracle-based logs, add about 10% to 15% for overhead.

3.1.4.3 Disk Requirements for File-based Numeric Logs

For file-based logs, the actual disk space may be greater than the calculated capacity if the log is very small. For TYPE1 (24-byte) logs, the minimum capacity allocated on disk is 84 entries. For TYPE2 (4-byte) logs, the minimum space allocated on disk is 500 entries.

Each TYPE1 log requires about 24 bytes per entry. To calculate exact file size for TYPE1:

- #data_blocks = (#entries+83)/84
- #index_blocks = (#data_blocks+253)/254
- file size = #data_blocks+#index_blocks)*2048

Each TYPE2 log requires about 4 bytes per entry. To calculate exact file size for TYPE2:

- #data_blocks = 1+ (#entries+499)/500
- #index_blocks = (#data_blocks+253)/254
- file size = #data_blocks+#index_blocks)*2048

For file-based storage, UNIX allocates 5% overhead per file.

3.1.5 Presentation (Trend), Engineering Units, and Alarm Attributes

The numeric log contains a number of attributes that are similar to attributes in the data source objects. Default values for these attributes are usually read directly from the data source. For most applications, you should not make any entries for these attributes through the History Builder for reasons described below.

3.1.5.1 Presentation Units

The Presentation Maximum and Presentation Minimum attributes support trend curve drawing. When a trend graph is requested, the AdvaCommand User Interface uses these two attributes to determine the upper and lower limits to use for the trend grid.

3.1.5.2 Engineering Units, and Alarm Attributes

There are a number of numeric log attributes that specify parameters related to engineering limits scaling, alarm detection, and device state identification. These attributes include:

- Engineering Units (Label)
- State Set - name of the set of state descriptors for device loops
- Engineering Maximum Value
- Engineering Minimum Value
- High High Limit
- High Limit
- Low Low Limit
- Low Limit

The engineering units and alarm values are present in the log for information purposes only. The log does not scale variables or check for alarms. The attributes can be read by an application program that needs this information about the data source.

3.1.5.3 Configuring the Presentation, Engineering Units and Alarm Attributes

It is recommended that you do not configure these attributes. Their values are determined by the data source. This allows you to have the same values for these attributes in the log as in the actual data source. The primary log acquires these attributes whenever it activates. If there are secondary logs in the log hierarchy, the values are also propagated to the secondary logs. Checks are made so that:

- used-entered values are not overwritten
- attributes are not collected if done previously. This is done to reduce the load when many logs are activated and shorten the time that History is busy storing attributes rather than entries. The flag indicating they have been collected can be cleared so that they are collected the next time the log is activated. This can be done using hsDBMaint. It can be done for all logs, all logs in a composite log, or a single log.
- attributes are not updated in the data base if identical to the previous values.

NOTE

If you tune one of these attributes in the data source object, you should:

- use hsDBMaint setStatus function to clear attrUpdated Bits. When these bits are cleared, History reads the attribute values from the data source when the log is activated.
- deactivate and then activate the associated log to ensure that the primary log receives the tuning change.

3.1.6 Data Retrieval Considerations

You can build applications to retrieve historical data using, AdvaCommand Trend Display, AdvaInform Reports, AdvaInform SQL*Connect, AdvaInform User API, and AdvaInform Display. This section describes how applications built with these software packages retrieve data from History. Refer to this section when you plan how to set up your log hierarchies. How you build your log hierarchies can have a significant impact on how data is retrieved from History, so it is important to have a good understanding of how each particular application retrieves historical data.

NOTE

Data that has been collected but not yet stored to disk by History (due to the sample blocking rate) is not available for retrieval.

3.1.6.1 When to Use Log Name or Access Name

There are two ways to specify which log to retrieve data from. The access name is the name of the data source that you assign to a log when you configure it in the History Builder. For instance, TC100,MEASURE or TC100,VALUE.

By virtue of the hierarchical log structure of AdvaInform History logs, there can be more than one log using the same data source. To distinguish multiple logs with the same data source, History automatically assigns a unique default log name to each log. The log name is derived from the access name, and has a prefix and suffix appended as shown in Figure 3-10. You can change the default log name when you configure the log in the History Builder.

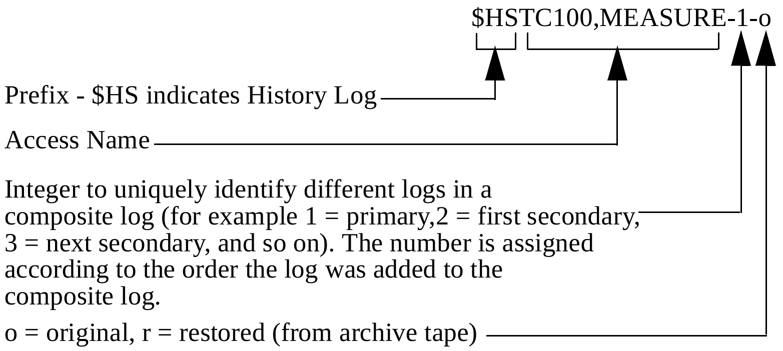

$HSTC100,MEASURE-1-o
Prefix - $HS indicates History Log
Access Name
Integer to uniquely identify different logs in a composite log (for example 1 = primary,2 = first secondary, 3 = next secondary, and so on). The number is assigned according to the order the log was added to the composite log.
o = original, r = restored (from archive tape)

Figure 3-10. Log Name

You can use the access name to specify which log to retrieve data from when you know the data source but do not know the log name. When you use the access name, History uses the seamless retrieval criteria (described later in this section) to select the most appropriate log in the data source's log hierarchy. Generally, unless other criteria are specified, seamless retrieval selects the log that has INSTANTANEOUS or AVERAGE as the calculation algorithm, and has a log period that provides the best coverage of the requested time period.

Use the log name when you do not want to rely on the seamless retrieval criteria to select the log, and you know the log name of the log that stores the required data. For instance, you can use the log name if you want the log that uses the STANDARD DEVIATION algorithm, and know that this data is stored in $HSTC100,MEASURE-3-o (or $HSTC100,VALUE-3-o).

3.1.6.2 Retrieval for AdvaCommand Trend Displays in MOD 300

The AdvaCommand software contains standard trend displays. All you have to do to configure these displays is to specify the variables you wish to see on them. You do this via the Group and Area definition phases of the AdvaCommand Environment Builder. AdvaCommand also has a trend element which allows you to put trend graphs on displays you define with the AdvaBuild Display Builder.

During runtime, trend data for CCF loops is collected at the controller and is stored locally in a CCF trend buffer at the controller. This buffer holds up to 300 samples. The CCF trend rate is configured on an individual basis for each loop via the Loop Definition Template. The trend rate determines the time period for the X axis. For instance, if the trend rate is 12 seconds (five samples per minute), the time period for the X Axis will be one hour (300 samples/5 samples per minute = 60 minutes).

The AdvaCommand trend display is built with data from the CCF trend buffer, unless a data request is entered to extend the X Axis back in time beyond the capacity of the buffer, or no CCF trend is configured. Such a request can be entered via the Modify X Axis dialog as described in the *AdvaCommand Basic Functions User's Guide*. In this case, the appropriate history log is used to build most of the trend, and the gap from the beginning of the log to present (due to delay caused by storage interval and sample blocking rate) is filled by the CCF trend buffer.

To extend the X axis, there must be one or more history logs configured for the CCF loop whose trend is being displayed. If the log name is used in the data request, History knows exactly which log to retrieve data from. If the access name is used, the seamless retrieval criteria is used to select the most appropriate log.

The following example illustrates how a CCF trend is built according to a data request entered via the Modify X Axis dialog.

A data request is entered in the Modify X Axis dialog for 8 hours worth of data from TC100,MEASURE (access name). A History log with a composite log hierarchy as shown in Figure 3-11 is configured in the History database for TC100,MEASURE.

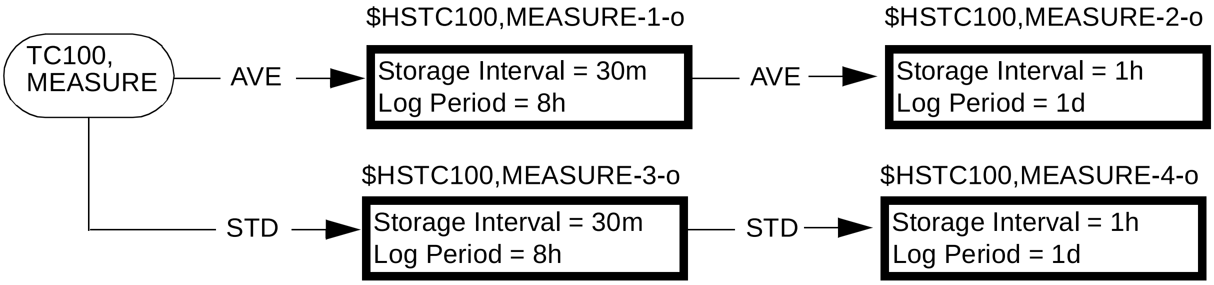


Figure 3-11. Example, Log Hierarchy

The log that most closely fits the data request is $HSTC100,MEASURE-1-o since the log period is eight hours and the calculation algorithm is AVERAGE. Since the storage interval is 30 minutes, there is the potential to lack as much as 30 minutes of the most current data, or possibly more since the gap is dependent on the blocking rate. This gap is filled by data from the CCF trend buffer. The trend display is shown in Figure 3-12.

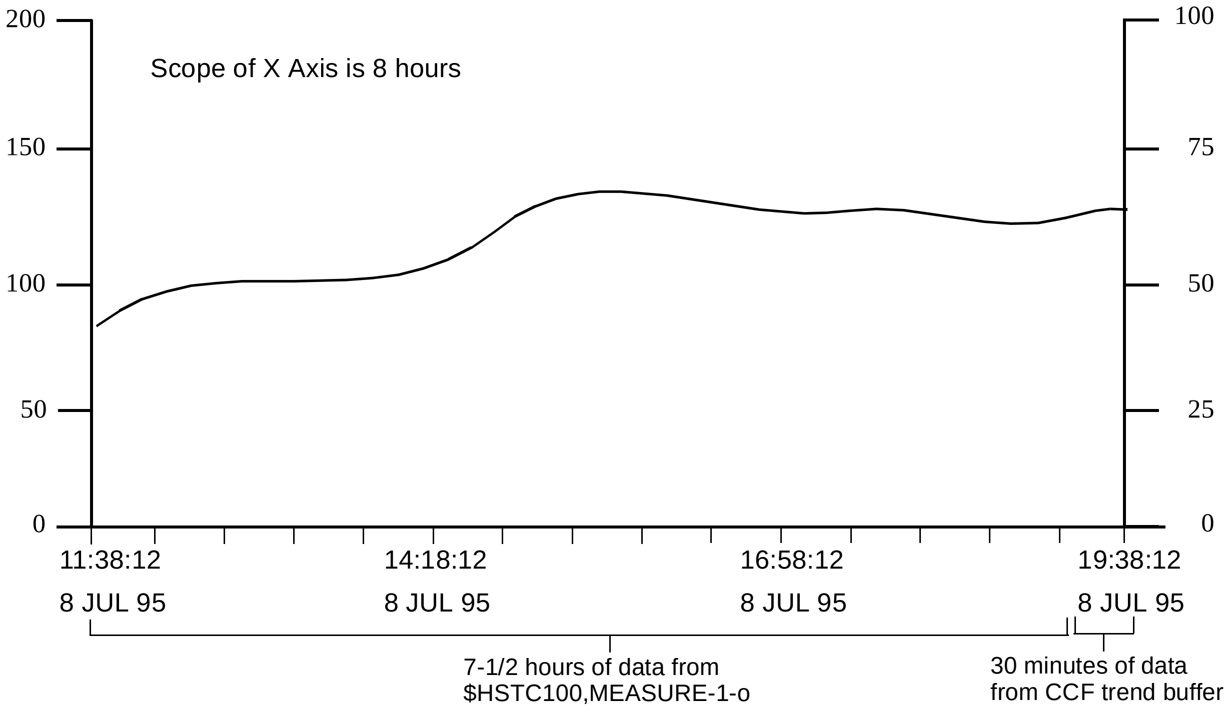


Figure 3-12. Example Trend Display

The trend display lets you scroll forward and backward in time. If you scroll backward in time and go beyond the limit of the history log that is currently being accessed, the seamless retrieval function will automatically switch to the most appropriate log that contains older data in the log hierarchy. Again, the selection criteria is according to the seamless retrieval function. For instance, if you use the scrolling function of the Modify X axis dialog to scroll back in time beyond the capacity of $HSTC100,MEASURE-1-o, the seamless retrieval function will automatically switch to $HSTC100,MEASURE-2-o which holds 24 hours worth of data.

NOTE

You can request data from a specific log in a composite hierarchy which would not normally be selected by the seamless retrieval criteria. For instance $HSTC100,MEASURE-3-o, would not normally be selected because it uses the STANDARD DEVIATION calculation algorithm. To force the selection of this log, enter the log name in the Trend Source field of the Modify Trend Traces dialog as described in the *AdvaCommand Basic Functions User's Guide*. You must use a period in place of the comma in the log name (for instance $HSTC100,MEASURE-3-o is entered as $HSTC100.MEASURE-3-o in the Trend Source field).

3.1.6.3 Retrieval for AdvaCommand Trend Displays in Master

The AdvaCommand User Interface contains standard trend display elements. You configure your trend displays via the AdvaBuild Display Builder. This section briefly describes how to configure your trend displays. A more detailed description is provided in the *AdvaCommand Basic Functions User's Guide*.

To build a trend display:

1. In the AdvaBuild Display Builder, select the option for making a new History trend display.
2. Select the attribute dialog (Tools/Display Attributes).
3. Select the Object names dialog and enter, ObjectName1, AttributeName1, and domain1.
4. Select the Display Element Attributes dialog and choose resolution and amount of values (TimeScope) to present.
5. Save the trend display.
6. Use the Display Installer to install the new trend display.
7. Select the display from the list of displays.

Define the Objectname1 attribute in terms of the access name of the composite log in History. Use of the access name provides for seamless retrieval.

If the access name is defined as ‘objectname,attribute’, you should set ObjectName1 to ‘objectname’, and AttributeName1 to ‘attribute’. This allows for seamless retrieval and the means for object selection within the trend display.

The ObjectName1 attribute can also be defined as the log name for the log that you want to display (for example, $HSobjectname,attribute-4-o). This may be useful if a certain log function calculation is to be displayed (for example, standard deviation).

3.1.6.4 Retrieval for SQL*Connect and AdvaInform Reports

To access data from a numeric log via AdvaInform SQL*Connect or AdvaInform Reports, you write a query against the NUMLOGVAL table which has the attributes shown in Table 3-4.

Table 3-4. NUMLOGVAL Attributes

Attribute	Data Type	Description
NAME	VARCHAR2(82)	Log name or Access Name
TIME	DATE	Date of the stored value
MS	NUMBER(10)	Millisecond time of the stored value
ENTRYVALUE	FLOAT(24)	The stored value
ENTRYSTATUS	NUMBER(10)	The status of the stored value. See Table 3-3.
OBJECTSTATUS	NUMBER(10)	The status of the object that provided the stored value.
RET_TYPE	VARCHAR2 (14)	Retrieval type. Possible values are: RAW - Actual sampled value INTERPOLATE - Interpolated value between samples
INTERP_ALG	VARCHAR2 (6)	Interpolation algorithm. Possible values are: AVE - Average MAX - Maximum MIN - Minimum INST - Instantaneous
SELECTION	VARCHAR2 (10)	When you use the Modify Numeric Value function (USER API), SELECTION can be used to specify current data or original data: CURRENT - Current (modified) value ORIGINAL - Value before modification
CALC_ALG	VARCHAR2 (14)	Used to specify the log whose calculation algorithm matches this specification: AVERAGE MAXIMUM MINIMUM INSTANTANEOUS SUM SUM_OF_SQRS STANDARD_DEV NUM_OF_VALUES WILDCARD
INTERVAL	NUMBER (10)	Interval for data retrieval in seconds. For instance to retrieve values at 1-hour intervals, enter 3600.

To query the object type you specify the log name (or access name) and the time period of the desired data. If you use access name, the log will be selected according to the seamless retrieval criteria described later in this section. To guarantee selection of a specific log in a composite hierarchy, you should use the log name.

Example using log name:

```
SELECT TO_CHAR (TIME, ‘DD/MM/YY HH:MI:SS’) time, ENTRYVALUE FROM
NUMLOGVAL
    WHERE NAME='$HSTC100,VALUE-1-o' AND
    TIME > TO_DATE('11 8:30:00', 'DD HH24:MI:SS')
AND
    TIME < TO_DATE('11 9:00:00', 'DD HH24:MI:SS')
```

Example using access name:

```
SELECT TO_CHAR (TIME, ‘DD/MM/YY HH:MI:SS’) time, ENTRYVALUE FROM
NUMLOGVAL
    WHERE NAME='TC100,VALUE' AND
    TIME > TO_DATE('11 8:30:00', 'DD HH24:MI:SS')
AND
    TIME < TO_DATE('11 9:00:00', 'DD HH24:MI:SS')
```

The to_char function in the ‘select’ clause of the query, returns the time in the format dd/mm/yy hh:mi:ss.

The ‘where’ clause uses the History software naming format. For example, if you build a primary log in History with an access name of TC100, MEASURE, the software assigns the log name $HSTC100,MEASURE-1-o to the log. You can use either naming convention (access name or log name) in the query.

The ‘where’ clause in the example specifies both the beginning and the end of the time interval which in this case is 8:30 to 9:00 of the 11th day of this month. If you do not specify a time interval, the most recent 500 values are returned. If you specify either the beginning or the end time of the interval, but not both, 500 values are returned.

The example query retrieves the entry values and their time stamps. The information is provided in columns:

```
        time               entryvalue
-------------------------------------------
11/10/93 08:30:00      150.8
11/10/93 08:30:10      151.0
11/10/93 08:30:20      150.9
11/10/93 08:30:30      150.8
and so on...
```

For further information regarding queries for History data, refer to the *AdvaInform Reports User’s Guide* and *AdvaInform SQL*Connect User’s Guide*.

NOTE

History configuration attributes are also available via SQL using the NUMLOGCONFIG view. Refer to the *AdvaInform Object Types Reference Manual* for details regarding this view.

3.1.6.5 Retrieval for AdvaInform User API

AdvaInform User API provides functions for retrieval of historical data. These functions allow you to specify the log based on either the log name, or any combination of the following criteria:

- Access Name
- Log Period
- Calculation Algorithm
- Retrieval Type

If you use the log name, History knows exactly which log to retrieve data from. If you use some combination of the access name, log period, calculation algorithm, or retrieval type, the seamless retrieval function will search the History database and select the log that most closely fits the specified criteria. The getHistData2 User API example located in the directory: `/opt/advant/UserAPI/examples` illustrates how a User API function can use these criteria to specify a log for retrieval.

To merge CCF and TTD data with History:

- Time Type must be PRESENT
- Time Justification must be ENDTIME
- Retrieval Type must be DISPLAY

If merging is done, the data will be returned in two responses. The first response is from History, and the second response is from CCF or TTD.

3.1.6.6 Criteria for selection for Seamless Retrieval

Seamless retrieval uses the following criteria to search the History database and select a log when a request for history data is issued:

1. Sort logs by percentage of the time period of the request that the log covers, and choose the set of logs that are closest in coverage.

   For instance, for a request for 24 hours, logs may be sorted as follows in descending order of preference: 24-hour logs and 168_hour logs, 8-hour logs, and 1-hour logs. Since the 24-hour logs and 168-hour logs cover 100% of the requested time period, they are chosen.

2. Search the set chosen in step 1, and find all logs that use the specified calculation algorithm. Logs that do not use the specified algorithm are dropped from consideration.

   If the algorithm is WILDCARD (no algorithm specified), then the algorithms have the following order of preference:

   a. AVERAGE or INSTANTANEOUS

      These algorithms have equal preference. The first one found in the search is selected, and logs with any other calculation are dropped from consideration.

   b. MAX

   c. MIN

   d. SUM

   e. SUM OF SQUARES

   f. STANDARD DEVIATION

   g. NUM_OF_VALUES

   At this point the set of logs still under consideration have the same log period (closest to percent of coverage of requested time period), and calculation algorithm.

3. This set is now searched for the specified retrieval type. One log is selected according to retrieval type as follows:

   a. If the retrieval type is RAW, one log is selected in the following order of preference:

      1) Pick the log whose storage interval is equal to the time between points in request.

      2) Pick the log whose storage interval is greater than and closest to the time between points (so request can be completed in one response).

      3) Pick the log whose storage interval is less than and closest to the time between points (more points exist than buffer can hold, MORE_DATA status is returned).

   b. If the retrieval type is INTERPOLATED, one log is selected in the following order of preference:

      1) Pick the log whose storage interval is less than and closest to the time between points (so interpolation is more accurate).

      2) Pick the log whose storage interval is equal to the time between points in request.

      3) Pick the log whose storage interval is greater than and closest to the time between points.

c. If retrieval type is DISPLAY, one log is selected according to retrieval type as follows:

1) Pick the log whose storage interval is equal to the time between points in request. The request is changed to RAW unless the log uses Deadband compaction, in which case the request is changed to INTERPOLATE/AVE.

2) Pick the log whose storage interval is greater than and closest to the time between points. The request is changed to INTERPOLATE/AVE.

3) Pick the log whose storage interval is less than and closest to the time between points. The request is changed to INTERPOLATE/AVE.

In addition to the seamless algorithm applied for retrieval requests, the following criteria have been added to better handle selection of a log for a request by access name, when logs are in a dual configuration:

- Uptime of node where log exists
- local is chosen over remote (local should always be faster than remote)
- sequence number of log. (with all other conditions equal, the lowest sequence numbered log is used. Sequence number is the ‘-1-o’ or ‘-2-o’ attached to the generated log name.

3.1.6.7 Interpolation

History usually samples the values of numeric variables at set intervals. When data is requested at the interval it was stored, the software returns raw data. If data is requested at a different interval, the software returns an interpolated value.

Figure 3-13 shows an example where data is saved every 10 seconds.

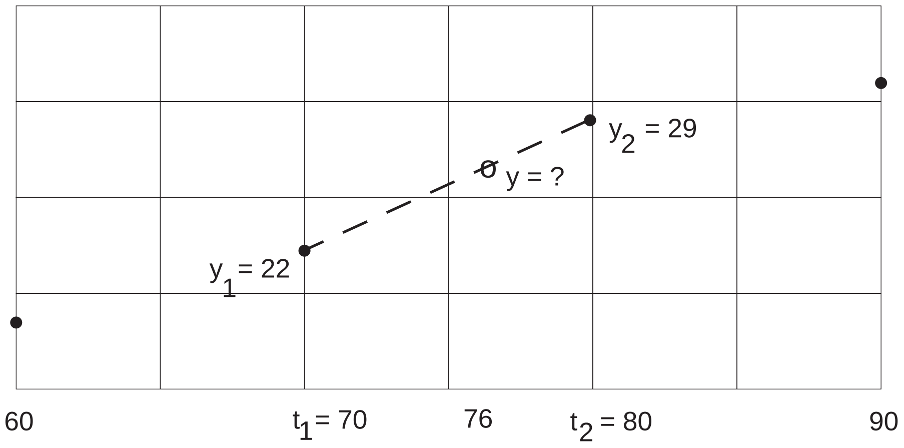


Figure 3-13. Numerical Data Retrieval by Interpolation

If the value is requested at second 70, the stored value 22 is returned. If the value is requested at second 80, the stored value 29 is returned. If the value is requested at second 76 (in between intervals where the value is stored), the stored values for seconds 70 and 80 are used to perform a mathematical interpolation. In this case, the value 26.2 is returned.

3.1.7 Archive Considerations

Timed archive of History data is supported using Archive Groups. An archive group contains a list of History logs and defines the frequency that archiving will be done for those logs (archive interval). One archive of the logs in an archive group is called an archive entry.

It is very expensive (in terms of tape used) to archive frequently. It is more economical to maximize the time between archives. However, there are a couple of factors that restrict the size of an individual archive:

- Never have an archive interval that is greater the 1/4 of a log's log time period. For example, if the log time period is 4 wk (four weeks), you should archive at least once a week. This restriction allows a number of archives to fail (due to tape full or drive problems) and not lose any data.
- For Numeric Logs use the following guidelines:
  - 1000 logs per archive group
  - 10,000 entries per log per archive

  For example, for a storage interval of 30s, that is 2 entries per minute, 10,000 entries in 5,000 minutes or 83 hours for the archive interval.
- For Message Log: max of 50,000 entries in a single archive

The space used on tape is the same regardless of the storage type of the logs.

3.1.8 History Log Attributes

Table 3-5 describes attributes common to all logs except PDLs. These attributes usually control basic functions such as identifying the log and turning the log on and off. Most attributes can be defined initially when you build the log and then changed on-line. If an attribute can not updated on-line, it is described in the attribute description.

Table 3-6 describes the attributes that are specific to numeric logs.

Table 3-5. Attributes Common to all Logs

Name	Description	Valid Entries/States
Identification		
Log Name	The log name must be unique system wide. It is derived from the access name that you enter via the Composite Log window. History assigns a default log name by adding a prefix $HS and a suffix -*n*-o to the access name. *n* is an integer that uniquely identifies different logs in a composite log. The number is assigned by History according to the order logs are added to the composite log. For example, a primary log with access name TC100,MEASURE, has the log name $HSTC100,MEASURE-1-o. The next (secondary) log will have the log name $HSTC100,MEASURE-2-o, and so on. The last character in the log name indicates whether the log is original (from disk) or restored from tape: o = original, r = restored	A text string, 80 characters maximum. All characters are allowed except!?: * and%.
Description	An optional text string for documentation purposes only. It does not have to be unique system wide. For numeric logs, the Description is not filled in by the user. It is derived from the data source.	A text string, 40 characters maximum. All characters allowed.
Access Name	The name you assign the log for retrieval. Defaults to the name entered in the composite log window, and is typically the same as the data source. All logs in a composite log have the same access name. You can use either the access name or log name to retrieve the data in a log.	A text string, 65 characters maximum. All characters allowed except: ! ? : * % - $. (period) is allowed, but recommended that you do not use unless absolutely necessary When the access name includes a database attribute, enter the attribute in UPPERCASE letters.
Log Group	Log group that this log belongs to. The log cannot be reassigned to a new log group.	Log group name (32 characters max.)
Log Set	Log set that this log belongs to. The log can be reassigned to a new log set.	Log set name (32 characters max.)
Archive Group	Archive Group containing this log. The log can be reassigned to a new archive group.	Archive group name (32 characters max.)

Table 3-5. Attributes Common to all Logs

Name	Description	Valid Entries/States
Collection		
Log State	Current state of the log. Individual logs can be activated or de-activated via the **Log** menu in the Composite Log Window.To activate a composite log, activate each of the individual logs in its hierarchy. You can also activate and de-activate multiple logs via log groups or log sets. When you activate a log group or a log set, the Log State attributes of all logs in the group or set go to Active.	I**NACTIVE** - not storing data (default) ACTIVE - storing data PENDING - log has been activated and is waiting to start collecting data. The time when a log is scheduled to start collecting data is referred to as the log's *alignment time*. Alignment time is the intersection of the log's activation time (see Activation Time below) and its storage interval. For instance if the activation time is Jun-30-95 10:00:00, and the storage interval is15 s, the log will store data at 00, 15, 30, and 45 seconds of each minute starting at 10:00:00 on June 30, 1995. In this case, this is the alignment time. If the activation time for a log has already passed, it will be aligned based on the storage interval. For example, a log with a storage interval of 1m will store data at the beginning of each minute. So its alignment time is the beginning of the next minute. The log stays in PENDING state until its alignment time. For the log whose storage interval is 1 minute, it will be PENDING until the start of the next minute. NOTE: Secondary logs go to active state immediately; however, they do not start collecting and storing data until the primary log becomes active.
Log Start-up State	Log state upon starting or restarting the node.	**START_INACTIVE** (default) START_ACTIVE START_LAST_STATE - restart in state log had when node shut down
Activation Time	The earliest allowable system time that the log can become active. If the specified activation time has already past, the log will begin to store data at the next storage interval. If the activation time is in the future, the log will go to PENDING state until the activation time arrives.	**JAN-01-70 12:00:00** (default) (The order of the month, day, and year and the abbreviation of the month is standardized based upon the language you are using.)

Table 3-5. Attributes Common to all Logs

Name	Description	Valid Entries/States
Size		
Log Period	Period of time that data is kept on disk for a log. This value is initially defined by the log group. It can be adjusted on an individual log basis. When you use compaction, the log period is effectively increased so that you store the same number of samples (as determined by the log capacity) over a longer period of time.	Integer value ≤ 4095 with time unit indicator: w or W= week, d or D= day, h or H= hour, m or M=minute, s or S= second. maximum = 4095w
Log Capacity	Maximum number of log entries for a log. If a log is full, new entries replace the oldest entries. For Asynchronous collection you must enter a value. For Synchronous collection log capacity is calculated as follows. Log Capacity = (Log Period/ Storage Interval). See Section 3.1.3, Data Collection Considerations for information on Log Period.	An integer value. The value must be at least equal to the calculated value for synchronous logs. (If you enter a value for Log Capacity and it is not ≥ the calculated value, then it is automatically updated to the calculated value.)

.

Table 3-6. Numeric Log Attributes

Name	Description	Valid Entries/States
Common Attributes: Numeric logs have the common attributes described in Table 3-5**.**		
Collection		
Collection Type	Determines whether History collects data (API), or whether data is sent to History by a user application (USER_SUPPLIED).	**API** - Values collected by History (default) USER_SUPPLIED - This mode allows a log to be written to like an asynchronous log, but have the same functionality as a normal synchronous log. SQL - Not supported in this release.
Collection Mode	Determines whether the data that History collects is periodic in nature (sampled at specific time intervals) or not. Collection mode specification is acquired from the Log group. Only Primary Logs can be Asynchronous.	**SYNCHRONOUS** - periodic (default) ASYNCHRONOUS - data sent when it occurs

Table 3-6. Numeric Log Attributes

Name	Description	Valid Entries/States
Storage Type	Method for storing log entries.	**TYPE1** - File-based storage, 24 bytes per entry. Full status information for each entry. TYPE2 - File-based storage, 4 bytes per entry. Subset of status information for each entry. ORACLE - Log entries stored in Oracle tables. Asynchronous logs must use this storage type.
Data Source	The process object or other source that data is collected from. This is used for logs whose collection mode is Synchronous and collection type is API.	The data source for the primary log is specified by the access name. When collecting data from TTD, the data source must be changed to the name of the TTD (for example TTD_10_72_1_41). Secondary log's data source attribute defaults to the object name of the predecessor log. (64 characters max.)
Sample Interval	The rate at which data is sampled from a data source. It is valid only for synchronous collection. If data is not received within 50% of the interval prior to, or 100% of the interval after the expected time, the entry value is marked with status NO_DATA. See Figure 3-2.	An Integer value with a lower or upper case time unit indicator: w (week), d (day), h (hour), m (minute), or s (second). ex. 5s or 5S, 15m or 15M Maximum = 4095w NOTE: Monthly is NOT a legal sample interval.

Table 3-6. Numeric Log Attributes

Name	Description	Valid Entries/States
Storage Interval	A time interval at which data is stored in the log, except when deadband is active. This value is acquired from the log group, and can not be changed. A special storage interval is monthly. This causes a value to be stored at the end of the month. The timestamp of a monthly log depends upon the storage interval and alignment of its data source. If the data source is a daily log which is aligned to 06:00, then the monthly log values will have a time of 06:00. If the data source is hourly or less, then the monthly log will be around midnight (depending on the alignment of the data source). For example, an hourly log at 30m after the hour will produce a monthly value at 00:30. If the data source is an 'x'h log aligned to 06:00 (for example, 8h log with values stored at 06:00, 14:00, 22:00, and so on), then the monthly log will have a time of 06:00	An Integer value with a lower or upper case time unit indicator: w (week), d (day), h (hour), m (minute), or s (second). Must be a multiple of the sample interval. ex. 10s or 10S Maximum = 4095w Values can also be stored at the end of the month. Enter the word monthly in the storage interval field in log groups.
Sample Blocking Rate	The amount of time that data is buffered in collection, before it is sent to storage. The sample blocking rate is only used in primary logs. If the data source is TTD, this is the rate at which data is collected from the TTD. Sample blocking rate is NOT used under the following conditions: - if the storage interval is Monthly - if the collection type is USER_SUPPLIED. In this case the values are stored as they are sent to History.	An Integer value with a lower or upper case time unit indicator: w (week), d (day), h (hour), m (minute), or s (second). Must be a multiple of the storage interval. ex. 10h or 10H Maximum = 4095w
Calculation		
Calculation Algorithm	Type of calculation performed on the data before it is stored.	AVERAGE MAXIMUM MINIMUM INSTANTANEOUS - no calculation (default) SUM SUM_OF_ SQRS STANDARD DEVIATION NUM_OF_ VALUES

Table 3-6. Numeric Log Attributes

Name	Description	Valid Entries/States
Number of Samples for Calculation	The number of samples needed for a calculation. If collection is Synchronous, then this value is derived from the Storage Interval/Sample Interval. If collection is Asynchronous, it is not used. This is a read-only attribute.	An Integer value
Bad Data Quality Limit	When the percentage of samples with bad data quality or no data status used in a calculation exceeds the Bad Data Quality Limit, the data quality of the calculated result is bad. For example, 20% means that 2 out of 10 data samples can be bad. This attribute can be used to avoid having data stored with BAD_DATA quality. When sampling remote objects, the interval between values may vary depending upon load in source node. In this case the sampled data will have status BAD_DATA. Use of this attribute allows some of the sampled values to be bad and still get a stored value with ok status. For example, set sample interval to 3s storage interval to 30s and make average. Set Bad Data Quality to 30%.	Percent ex. 10.500 default 0
Compaction		
Compaction Deadband	Minimum change between a sample and previously stored value that causes the new sample to be stored.	Percent of engineering units range (Range Max. - Range Min.) For example, 1 means 1% of range.
Deadband Storage Interval	When deadband is active, this is the maximum time that can elapse between storing entries. This assures that some data is stored, even if the process is stable.	An Integer value with a lower or upper case time unit indicator: w (week), d (day), h (hour), m (minute), or s (second). Must be a multiple of the storage interval. ex. 1s or 1S
Compaction Ratio	The number of entries sampled for each entry stored. Filled in by running hsDBMaint. See Section 5.6.5, Update Deadband Ratio On-line.	Read only.
Est_Log_Time_Per	Time between the oldest and newest stored values for the log. This is filled in by running hsDBMaint. See Section 5.6.5, Update Deadband Ratio On-line.	Read only.

Table 3-6. Numeric Log Attributes

Name	Description	Valid Entries/States
Presentation It is generally recommended that these attributes not be configured by you. Refer to Section 3.1.5, Presentation (Trend), Engineering Units, and Alarm Attributes for details.		
Presentation Maximum Value	The high value in setting the range for a trend grid. It is obtained from the data source when the log activates.	Value in engineering units. Three decimal places are allowed, and the maximum allowed value is 99999999
Presentation Minimum Value	The low value in setting the range for a trend grid. It is obtained from the data source when the log activates.	Value in engineering units. Three decimal places are allowed, and the minimum allowed value is -9999999
Engineering Units		
Engineering Units	Label for the value. Is obtained from the data source when the log activates.	A text string, 10 characters maximum.
State Set	Identifies the descriptor set for a device loop. It is obtained from the data source when the log activates.	Name of descriptor set
Engineering Maximum Value	High value of engineering range. It is obtained from the data source when the log activates.	Value in engineering units. Up to three decimal places. Max. value is 99999999
Engineering Minimum Value	Low value of engineering range. It is obtained from the data source when the log activates.	Value in engineering units. Up to three decimal places. Max. value is 99999999
Alarms		
High High Limit	Trip point for second high alarm. It is obtained from the data source when the log activates.	Value in engineering units. Up to three decimal places. Max. value is 99999999
High Limit	Trip point for first high alarm. It is obtained from the data source when the log activates.	Value in engineering units. Up to three decimal places. Max. value is 99999999
Low Limit	Trip point for first low alarm. It is obtained from the data source when the log activates.	Value in engineering units. Up to three decimal places. Max. value is 99999999
Low Low Limit	Trip point for second low alarm. It is obtained from the data source when the log activates.	Value in engineering units. Up to three decimal places. Max. value is 99999999

3.2 Capacity and Performance

3.2.1 CPU Performance Limitations

The following are general guidelines for capacity and performance.

3.2.1.1 Fastest Sample Rate

The fastest sample rate supported is 1 second periodic data. However, during periods when system load is high, one-second subscription to local or remote objects are likely to be interrupted. For example, if an AdvaCommand graphic display requests 16 History trends simultaneously, this spike in usage may cause the collection time of a sample to drift. As a result, NO_DATA entries will be inserted in the log when these spikes occur. Therefore, the fastest practical sample rate is based on the History database configuration, and use of History node resources by History and other applications. Refer to the *AdvaInform History Release Notes* for further information on frequency of NO_DATA entries.

3.2.1.2 Samples per Second

In MOD 300, the maximum number of samples that can be collected per second is 100 nominal.

For TTD applications (Master) the maximum number of samples for TTD is 415 per second; however, the system can not handle more than 4.3 TTD requests per second. Use the blocking rate to adjust this. You can use the stagger function in DBMaint to show the average number of requests to TTD per second. See Section 5.6.14, Stagger Collection and Storage for details.

3.2.1.3 Data Storage Per Second

The figures in this section are nominal. Overall system configuration will affect the performance. For instance, performance may be better if fewer logs are configured, or performance may be worse if many logs are configured. When calculating these figures, the sample blocking rate was used to optimize storage Refer to Section 3.1.3, Data Collection Considerations for explanation on how to use sample blocking rate to optimize storage. Also, the Stagger Collection function in the hsDBMaint utility provides a convenient means for adjusting blocking rate and activation time on-the-fly. Refer to Section 5.6.14, Stagger Collection and Storage.

Also, other AdvaInform applications should be considered.

The following figures are based on file-based storage for 10,000 logs and a sample blocking rate of 5:1:

- for a 50MHz machine: 167/second
- for a 64MHz machine: 250/second
- for a 99MHz machine: 333/second

3.2.1.4 Maximum Log Size

Log capacity and number of logs determine tablespace and disk requirements. For numeric logs with ORACLE storage type, the practical limit is 100,000 entries per log. For numeric logs with TYPE1 or TYPE2 storage type, the hard limit is 1,000,000 entries per log. This should be sufficient for virtually any application.

3.2.1.5 Log Group, Log Set, and Composite Log Size

- Number of logs in a log group: 1,000
- Number of logs in a log set: 1,000
- Number of logs in an archive group: 1,000
- Number of logs in a composite log: 16

3.2.1.6 Number of Primary Logs

The maximum number of primary logs is a function of sample rate. When data compaction is used, the estimated compaction ratio and deadband storage interval can be used to determine storages/minute. The practical limit is:

- MOD 300: 4,000
- Master: 8,000.

3.2.1.7 Number of Log Objects

The practical limit is 85,000. When you count the number of objects, count the composite log structure as one object, and each log within the composite log as one additional object. For instance:

- a primary log with no secondary log counts as two objects
- a composite log with one primary log and two secondary logs counts as four objects

3.2.2 Disk Capacity

3.2.2.1 Default Oracle Tablespace

History divides data into 5 different Oracle tablespaces. The default sizes of these tablespaces are as follows:

- Inform_HS_RunTime - 80 MB
- HS_Indexes - 160 MB
- HS_Archive - 15 MB
- HS_Restored - 115 MB
- HS_Reports - 10 MB

A History mount point for file-based logs is automatically created. It allocates 200 MB.

These default sizes provide space for approximately:

- 1000 logs
- 4,000,000 log entries
- 200,000 restored log entries

If your system requires greater storage capacity, you can extend the size of the applicable tablespace(s) and allocate more disk space for file-based logs as described in Section 5.6.7, How to Extend Tablespace for History Files. Keep track of the number of runtime logs in your system, the number of potential runtime log entries, and the number of restore log entries so you can extend the sizes BEFORE you exceed any of the above limits.

Keep a record of the number of logs you configure in the system. To determine the potential number of log entries in your system, calculate the sum of the log capacity attribute values for each numeric log in your system. You can view the number of restored logs via the Number of Logs field in the Archive Logs window. Section 4.4.3.2, How to Archive Logs describes how to display a list of logs. In this procedure, choose the **Retrieve by Location** option, and choose **Restored** Log Class in the Retrieve Log by Location dialog box.

3.2.3 Compatibility

Table 3-7 indicates compatibility between AdvaInform History and TTD (Master).

Table 3-7. Compatibility Table for MasterPiece Logs

Functionality	Yes	No
AdvaInform History can collect data from TTD	X	
AdvaInform History uses data from TTD for presentation	X	
AdvaInform History supports configurations defined in TTD		X

Table 3-8 describes the interaction between AdvaInform History and CCF Trends (MOD 300).

Table 3-8. Compatibility Table for CCF Logs

Functionality	Yes	No
AdvaInform History collects data from CCF Trends		X
AdvaInform History uses data from CCF Trends for presentation	X	
AdvaInform History supports configurations defined in CCF Trends		X

3.3 Application Start-up

When you run History on an Advant Station with AdvaInform software, you access the History displays from the IMS menu. When you run History on an Advant Station with AdvaCommand software (MOD 300 only), access is via the external display access feature. In either case, after you gain access to History windows, the operation of all History functions is the same.

3.3.1 Accessing History on Advant Station w/ AdvaCommand (MOD 300)

To access History displays on an Advant Station with AdvaCommand software, External Display Access must be configured in the AdvaCommand environment to support History windows. This procedure is described in the *AdvaBuild Environment Builder User's Guide*. Instructions for navigating windows in AdvaCommand are provided in the *AdvaCommand Basic Functions User's Guide*.

3.3.2 Accessing History on an Advant Station w/AdvaInform

This section describes how to open the History Configuration window from the IMS menu. For instructions on how to open the IMS menu, refer to the *AdvaInform Basic Functions User's Guide*.

You access the History Builder through the **AdvaInform** menu on the IMS Menu, Figure 3-14.

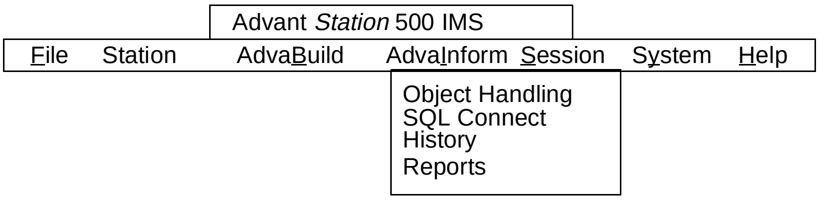


Figure 3-14. AdvaInform Menu Items

1. Choose **History** from the **AdvaInform** menu.

   This displays a submenu with two options (History Configuration and TTD-Configuration).

2. Choose **History Configuration** from the submenu. This displays the AdvaInform History window, Figure 3-15.

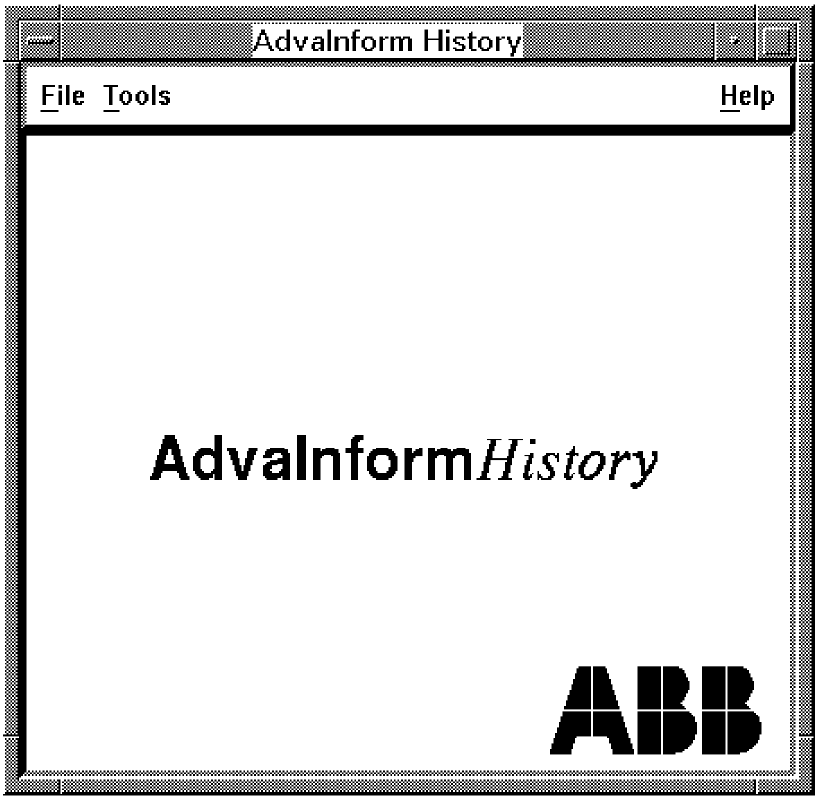


Figure 3-15. AdvaInform History Window

3.4 Tutorial

This section provides tutorials that take you step-by-step through the following basic History configuration applications:

- Store numeric values at a periodic rate
- Store numeric values in a composite log using log typicals and composite log typicals

These tutorials are intended to give you some quick hands-on experience and familiarize you with the tools and methods for History database configuration. Keep in mind that these tutorials are only an introduction. You should consult Section 3.5, Application Procedures and Section 3.6, Configuration/Application Menus for more comprehensive and detailed information on application procedures. Also, refer to Section 3.1, Design Considerations for information on how to define log attributes.

3.4.1 How to Begin a History Builder Session

Before you can begin configuring with the History Builder, you must log in to the application as described in Section 3.3, Application Start-up, and select the History Manager. Each History node has a History Manager. The History Manager keeps track of the data and operations for all History objects in a node.

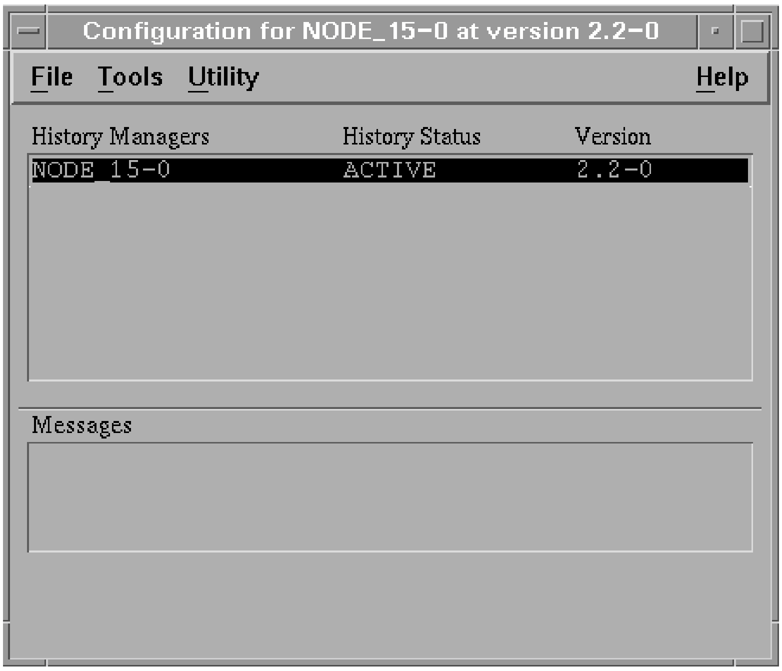
For MOD 300 applications, you can work with one or more History Managers at a time during a History Builder session; however, you must specify those History Managers before you begin.

To begin a History Builder session:

1. Log in to the History node and invoke the History Builder as described in Section 3.3, Application Start-up.

   This opens the AdvaInform History window.

2. Choose **Configuration** from the **Tools** menu. This opens the History Configuration window,Figure 3-16.


Configuration for NODE_15-0 at version 2.2-0

History Managers	History Status	Version
NODE_15-0	ACTIVE	2.2-0

Messages

Figure 3-16. History Configuration Window

   This window indicates the name of the History Manager for the local node (in this case NODE_15-0), and the current status of the History Manager.

3. Click once on the name of the History Manager to select it. If there is only one History Manager, as in Figure 3-16, it is automatically selected. In the example in Figure 3-16, since the History Manager on NODE_15-0 is selected, that is the History Manager that you can work with.

NOTE

These tutorials may use History object names that are not on your system. These names are only examples. You can substitute other names as you wish.

These tutorials assume that you have opened the History Configuration window, and have selected a History Manager.

3.4.2 Tutorial 1 - Storing Numeric Values at a Periodic Rate

This tutorial describes how to set up periodic data storage for process loops/objects. It is the most basic and straightforward application of History.

Tutorial 1 Application

Store data at a 25 second interval for two process objects/loops. This requires you to:

- build a log group that has a 25 second storage interval.
- use this log group and the Composite Log window to build the logs that store the data. When you are finished, you will have one log per loop as shown inFigure 3-17.

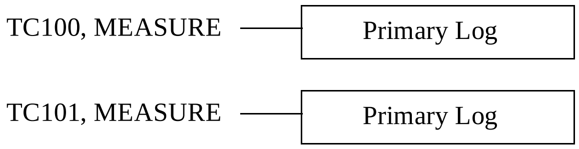


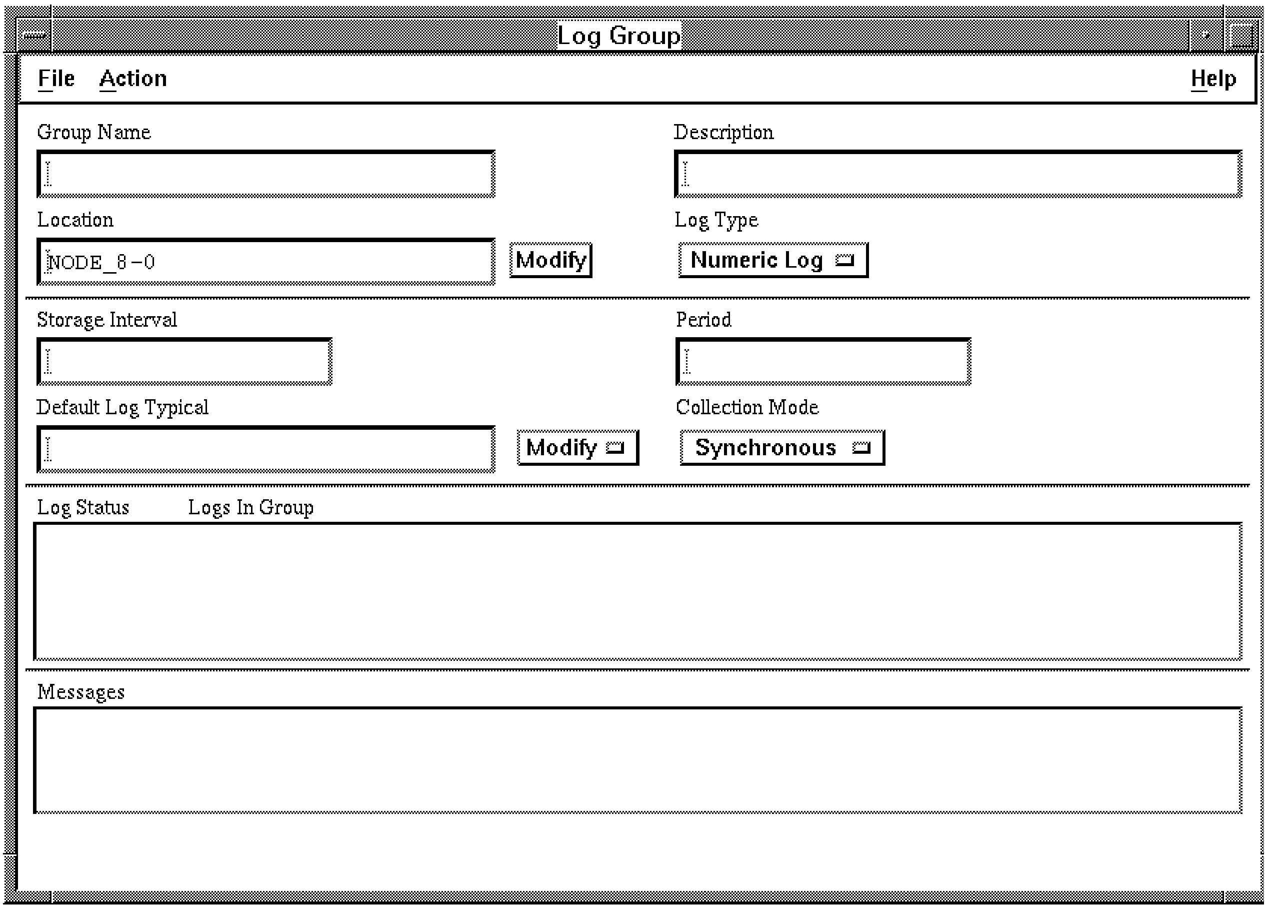
Figure 3-17. Tutorial 1 Application

To configure the History database for this application, follow the procedures described in the following sections.

3.4.2.1 Configure a Log Group

To configure the log group for this application:

1. Choose **Log Group** from the **Tools** menu of the History Configuration window. This opens the Log Group window, Figure 3-18.



Location: NODE_8-0

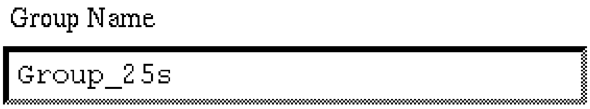
Figure 3-18. Log Group Window

When you open the Log Group window, if you selected the History Manager for the local node in the Configuration window (Section 3.4.1, How to Begin a History Builder Session), the Location defaults to the local node (in this case, NODE_8-0), the Log Type defaults to Numeric Log, and the Collection Mode defaults to Synchronous. These values are suitable for the application being configured in this tutorial.

Group Name, Storage Interval, and Log Period must be configured as described in the following steps.

2. Enter the log group name in the Group Name field, Figure 3-19.

First click inside the Group Name field, and then type the Log Group name.

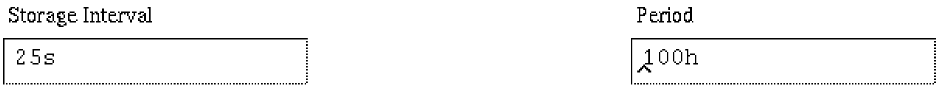


Group Name

Group_25s

Figure 3-19. Entering Log Group Name

3. Enter **`25s`** in the Storage Interval field and enter **`100h`** in the Period Field, Figure 3-20.



Storage Interval

25s

Period

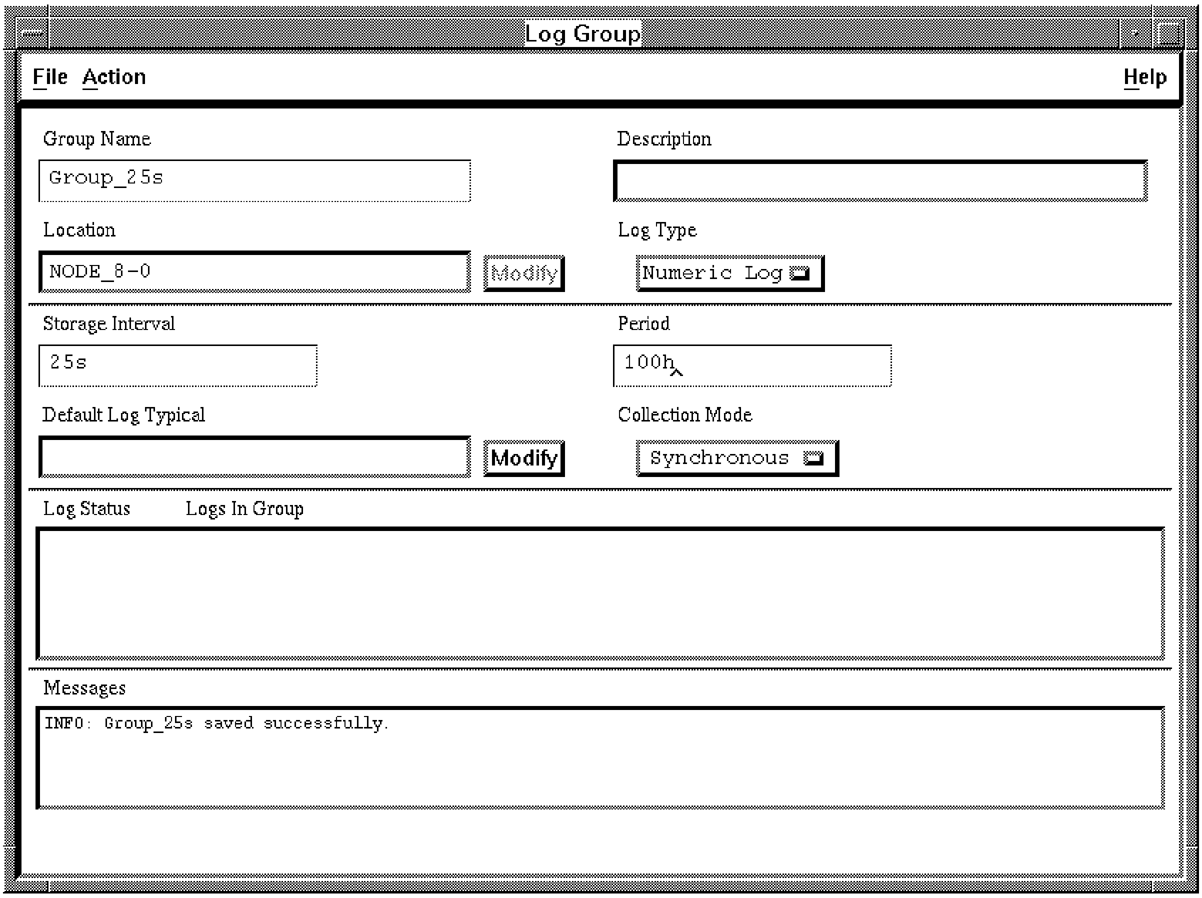
100h

Figure 3-20. Entering the Storage Interval and Period

Data will be stored every 25 seconds for all logs in this group. The period is the minimum time span of the log before data starts wrapping around. 100h is 100 hours.

4. Leave all other fields at their default values. Note that the log group description is optional.
5. To save the configuration, choose **Save** from the **File** menu in the Log Group window.

The completed log group is shown in Figure 3-21. A message indicating the save was successful is displayed in the Messages box on the bottom of the Log Group window.



Group Name: Group_25s
Description:
Location: NODE_8-0
Log Type: Numeric Log
Storage Interval: 25s
Period: 100h
Default Log Typical:
Collection Mode: Synchronous
Log Status
Logs In Group
Messages

```
INFO: Group_25s saved successfully.
```

Figure 3-21. Group_25s Completed and Saved

6. Choose **Exit** from the **File** menu to close the Log Group window.

3.4.2.2 Build a Basic Composite Log

Now build logs for log group Group_25s via the Composite Log window as follows:

1. Choose **Composite Log** from the **Tools** menu in the History Configuration window.

This opens the Composite Log window, Figure 3-22.

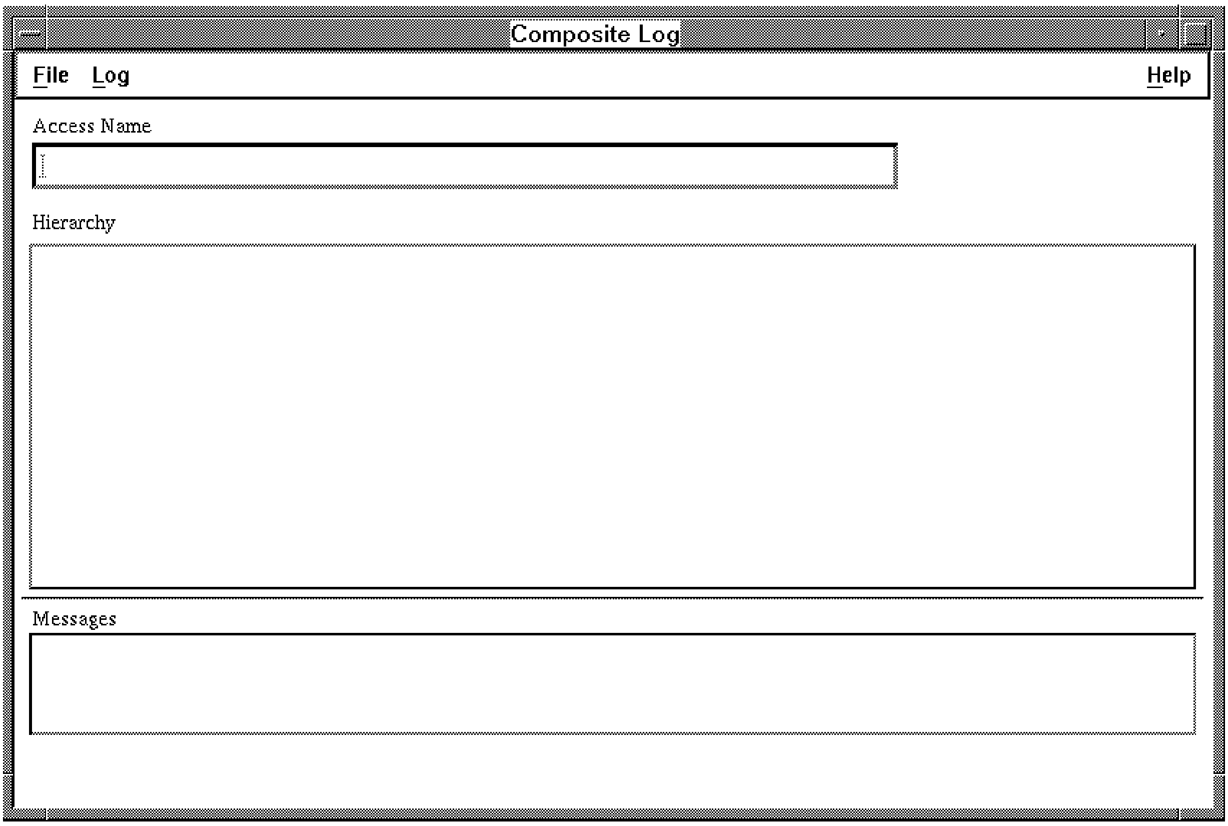
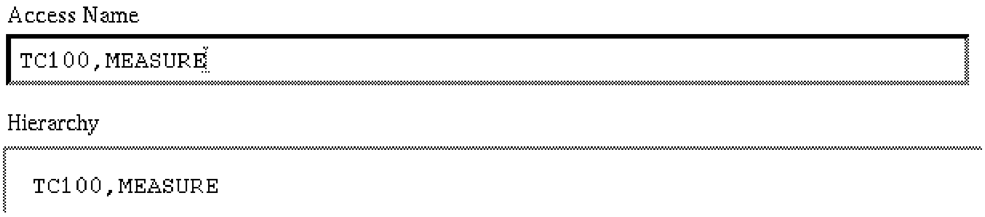


Figure 3-22. Composite Log Window

2. To specify the data source for the first log, enter the data source name in the Access Name field (for example TC100,MEASURE, or TC100,VALUE). Use all uppercase letters for the attribute name.

   As you enter a data source name, it is also entered in the Hierarchy box, Figure 3-23.



Access Name: TC100,MEASURE

Hierarchy: TC100,MEASURE

Figure 3-23. Specifying the Data Source

3. Specify Group_25s as the log group for the logs being built for this application. To do so:

   a. Click on the Access Name (TC100,MEASURE or TC100,VALUE)in the Hierarchy box to select it (selection indicated by highlighting, Figure 3-24.

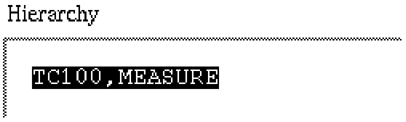


Figure 3-24. Selection Highlighted

   b. Then press and hold the right mouse button to display a pop-up menu, Figure 3-25.

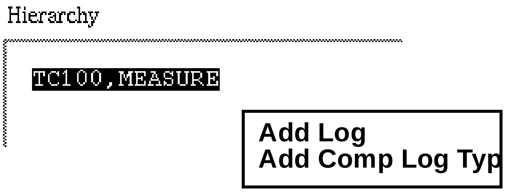
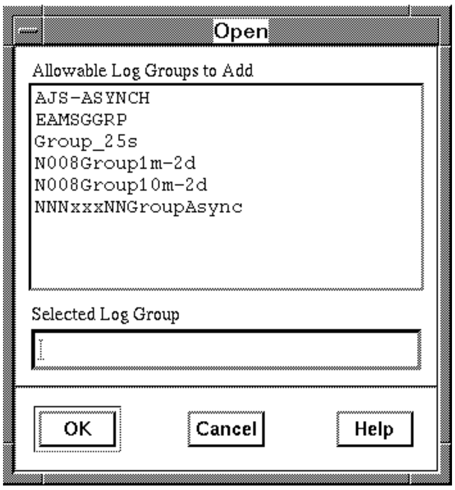


Figure 3-25. Pop-Up Menu for Adding Logs to Hierarchy

   c. Drag down to **Add Log** and release to select this option.

   This displays the allowable log groups to add, Figure 3-27.

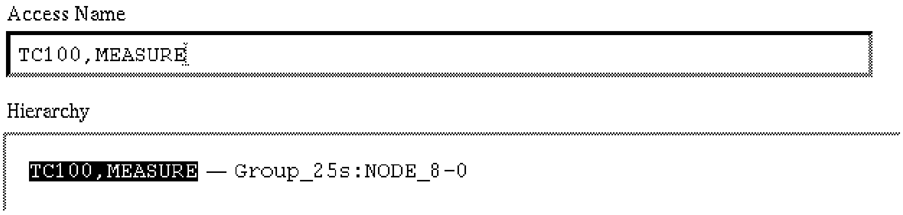


AJS-ASYNCH
EAMSGGRP
Group_25s
N008Group1m-2d
N008Group10m-2d
NNNxxxNNGroupAsync

Figure 3-26. Allowable Log Groups to Add

   d. Select the Group_25s from the Log Groups box, then click on **OK**.

This removes the dialog box. The selected group name is displayed in the Hierarchy box in the Composite Log window, Figure 3-27.



TC100,MEASURE

TC100,MEASURE — Group_25s:NODE_8-0

Figure 3-27. Selected Log Group Name Displayed in Hierarchy Box

3.4.2.3 Edit Log Attributes

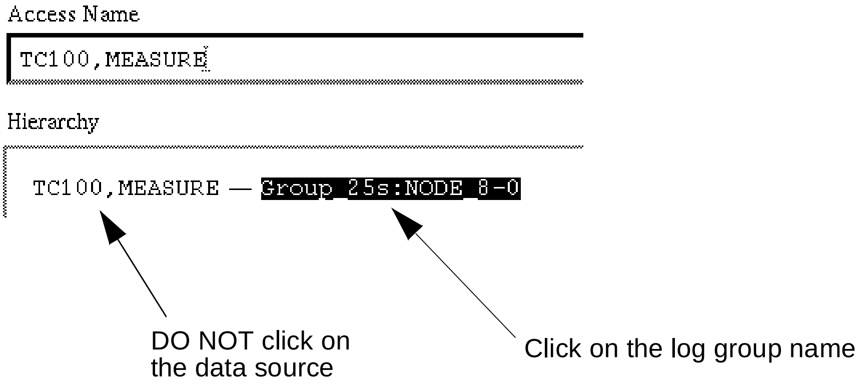
In this example, since no calculation algorithm is being applied, the sample interval must be equal to the storage interval that was specified for the log group. By default the sample interval is equal to the storage interval.

The sample blocking rate defaults to five times the storage interval. The blocking rate determines how long data will be buffered in data collection before it is stored on disk. Buffering data improves storage efficiency.

To edit the sample blocking rate for this log:

1. Click on the log in the Hierarchy box, Figure 3-28.

   DO NOT not click on `TC100,MEASURE or TC100,VALUE`, but rather, click on `Group_25s:NODE_8-0`. Use the log group name and NOT the data source to open the edit window for a log. This is because it is possible to have multiple logs connected to the same data source (via a composite log hierarchy). This is the only means to distinguish one log from another.

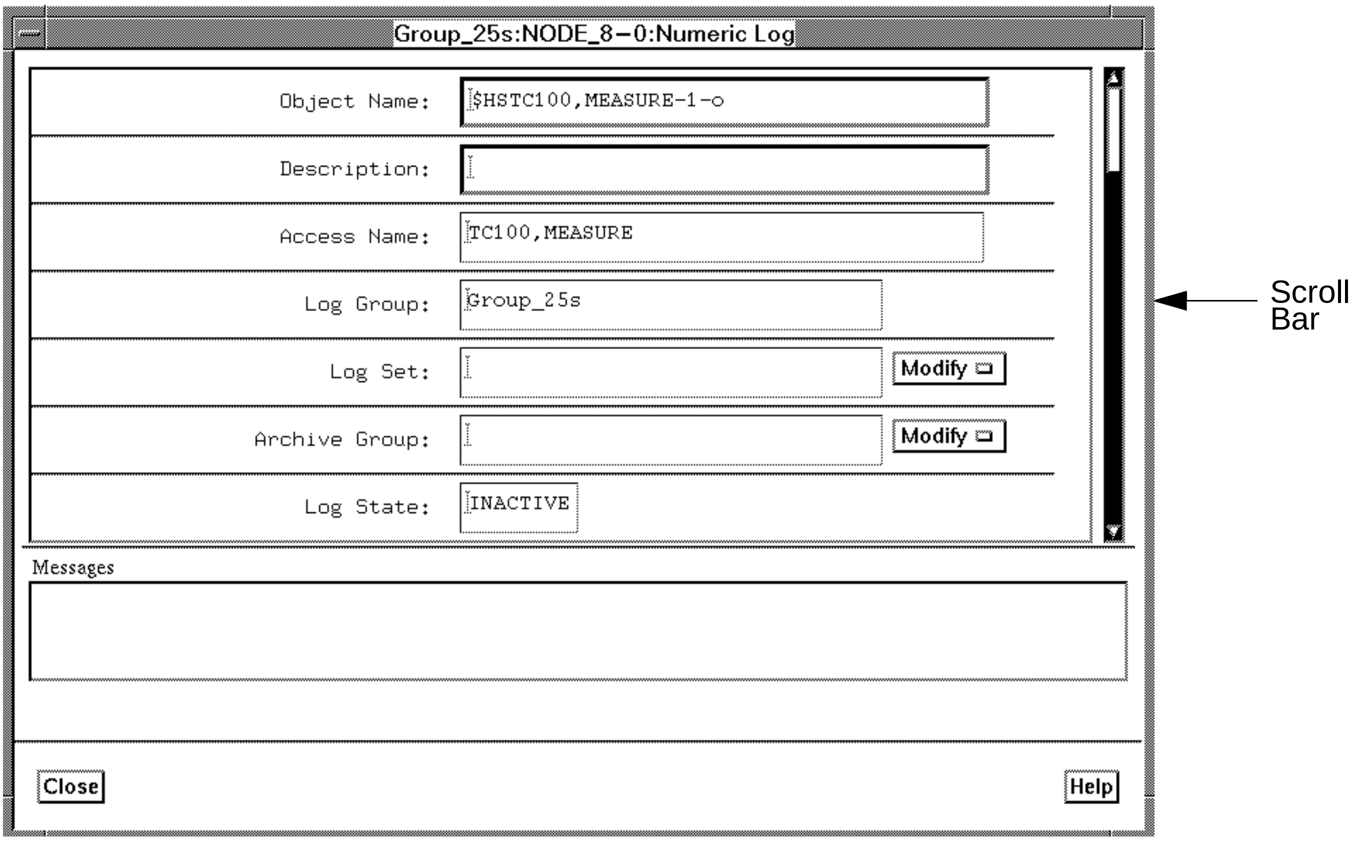


TC100,MEASURE

TC100,MEASURE — Group_25s:NODE_8-0

Figure 3-28. Selecting Log to Open

2. Choose **Open** from the **Log** menu (or press the right mouse button to get the same menu). This opens the Log Attribute Edit Window for the log, Figure 3-29. Use the scroll bar on the right side of this window to scroll up and down the list of attributes.



Group_25s:NODE_8-0:Numeric Log

Object Name: $HSTC100,MEASURE-1-o

Description:

Access Name: TC100,MEASURE

Log Group: Group_25s

Log Set:

Archive Group:

Log State: INACTIVE

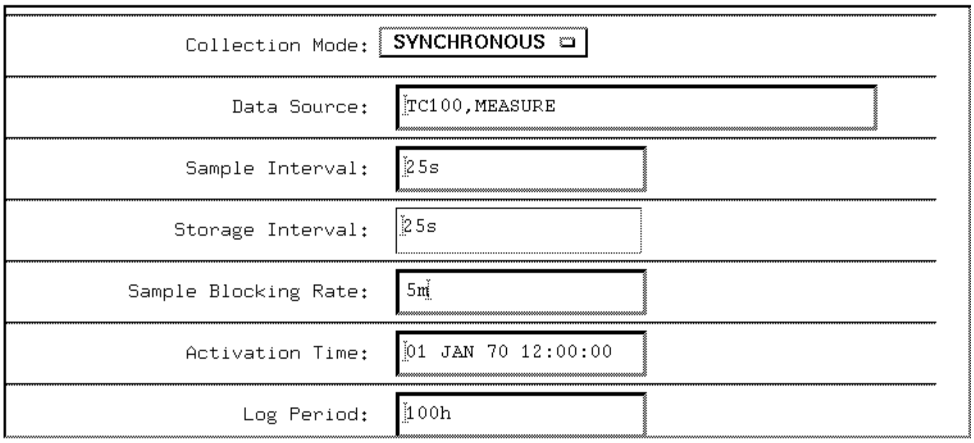
Figure 3-29. Log Attribute Edit Window for TC100,MEASURE Group_25s

History appends a prefix ($HS) and a suffix (-1-o) to the specified access name to make a log (object) name. The log name for TC100,MEASURE is `$HSTC100,MEASURE-1-o`. The log name parts are described below:

- The prefix $HS identifies the object as related to History.
- The first digit in the suffix (1), indicates the log's place in the composite log hierarchy. The number 1 indicates that this log is the first (primary) log in the hierarchy. This number is incremented for each secondary log that is added to the log hierarchy.
- the letter *o* indicates that this is an original log. If the log had been restored from an archive tape, this would be indicated by the letter *r*.

For History applications, the access name (TC100,MEASURE or TC100,VALUE) is sufficient for accessing Historical data for this object. Some application require the whole log name. Whether you require the whole log name or just the access name is described in the specific instructions for each application that can access History data.

3. Scroll to the Sample Blocking Rate attribute and enter **5m**, Figure 3-30

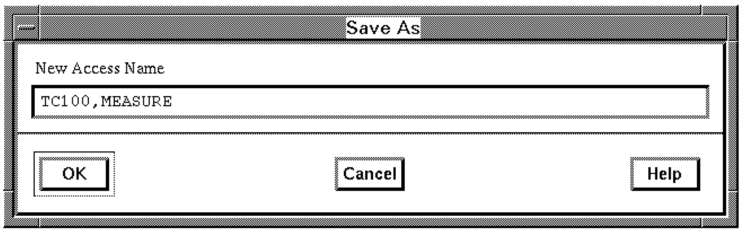


Collection Mode: SYNCHRONOUS
Data Source: TC100,MEASURE
Sample Interval: 25s
Storage Interval: 25s
Sample Blocking Rate: 5m
Activation Time: 01 JAN 70 12:00:00
Log Period: 100h

Figure 3-30. Changing the Sample Blocking Rate

4. To close the window, click on the **Close** button.
5. To save your work up to this point, choose **Save** from the **File** menu in the Composite log window.
6. Build the second log using the **Save As** item of the **File** Menu. To do so:
   a. Choose **Save As** from the **File** menu.

      This displays the Save As dialog box Figure 3-31. This dialog box is for specifying the data source for another log.



New Access Name: TC100,MEASURE

Figure 3-31. Save As Dialog Box

   b. Enter a new data source name (for example `TC101,MEASURE or TC101,VALUE`), then click on **OK**.

      This generates another log which is a copy of TC100,MEASURE or TC100,VALUE, but with a unique name and data source.

7. Choose **Exit** from the **File** menu on the Composite Log window to close the window.

This concludes the first tutorial.

3.4.3 Tutorial 2 - Building Log Hierarchies Using a Composite Log Typical

Tutorial 2 - Application

This tutorial demonstrates how to use log typicals and composite log typicals to build a log hierarchy as shown in Figure 3-32.

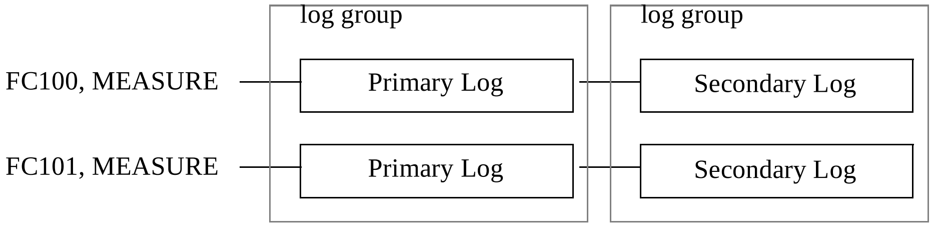


Figure 3-32. Log Hierarchy for Tutorial 2

There are four major parts to this procedure:

- Build two log typicals. One will be a template for the primary logs. The other will be a template for the secondary logs.
- Build two log groups. One group will be used to build the primary logs. The other group will be used to build the secondary logs.
- Build a composite log typical to serve as a template for the log hierarchy. The composite log typical specifies the hierarchy of the logs, that is to say, each composite log has a primary log in the primary group which provides input for a secondary log in the secondary group.
- Build the logs using the Composite Log Window.

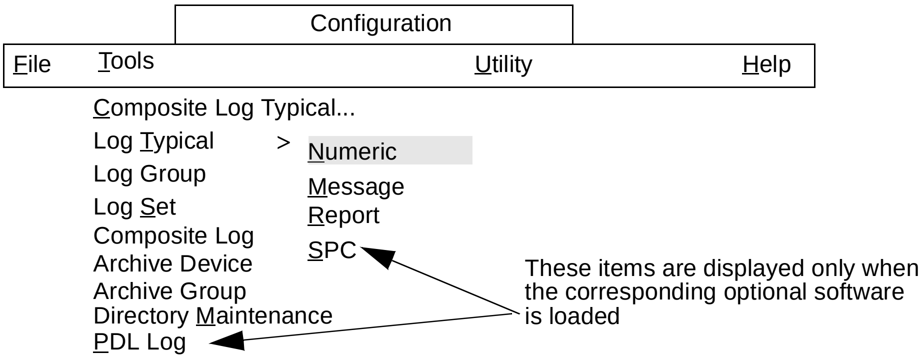
To build the application shown in Figure 3-32, follow the procedures described in the following sections.

3.4.3.1 Build Log Typicals

First build the log typicals as follows:

1. Choose **Log Typical** from the **Tools** menu.

This displays a submenu with a listing of log types, Figure 3-33.

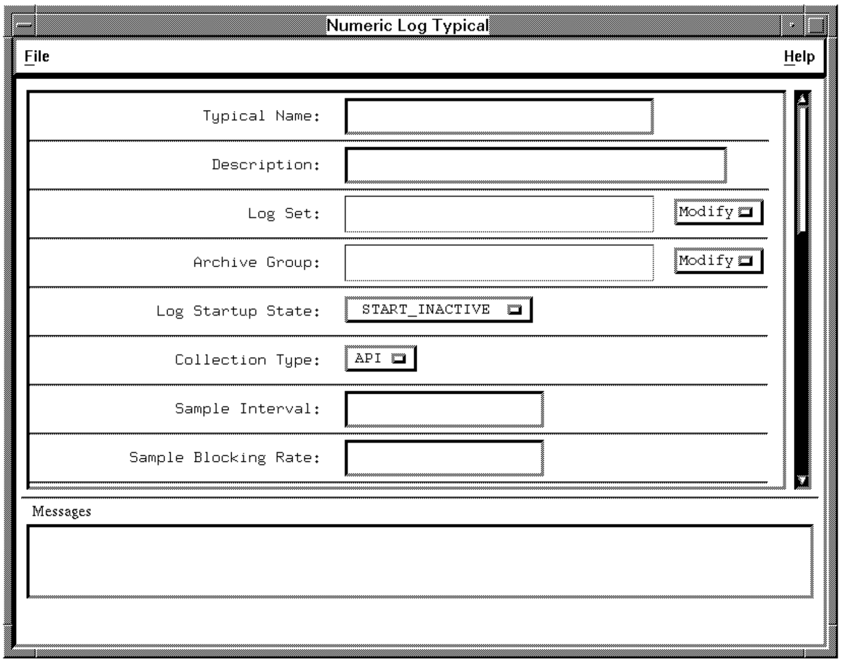


These items are displayed only when the corresponding optional software is loaded

Figure 3-33. Submenu for Selecting Log Type

2. Choose **Numeric**.

   This opens the Numeric Log Typical window, Figure 3-34.

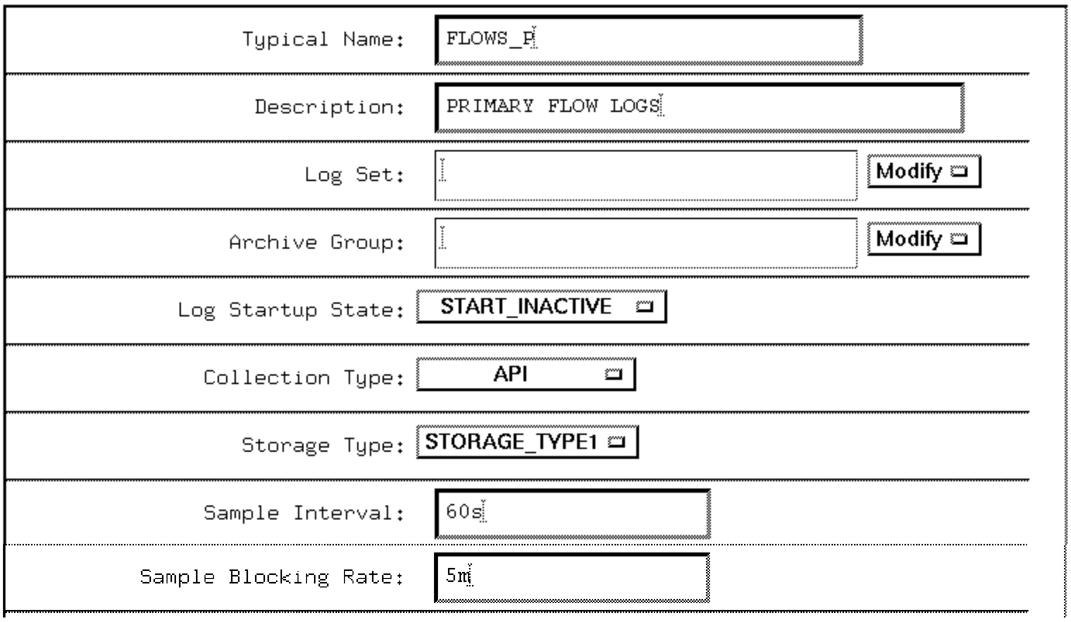


Log Startup State: START_INACTIVE

Collection Type: API

Figure 3-34. Numeric Log Typical Window

3. Build the log typical with the attribute values shown in Figure 3-35.



Typical Name: FLOWS_P

Description: PRIMARY FLOW LOGS

Log Set:

Archive Group:

Log Startup State: START_INACTIVE

Collection Type: API

Storage Type: STORAGE_TYPE1

Sample Interval: 60s

Sample Blocking Rate: 5m

Figure 3-35. FLOWS_P Log Typical Definition

For this tutorial, you will define the Typical Name, Description, and Sample Interval attributes. All other attributes can be left at their default values (or undefined).

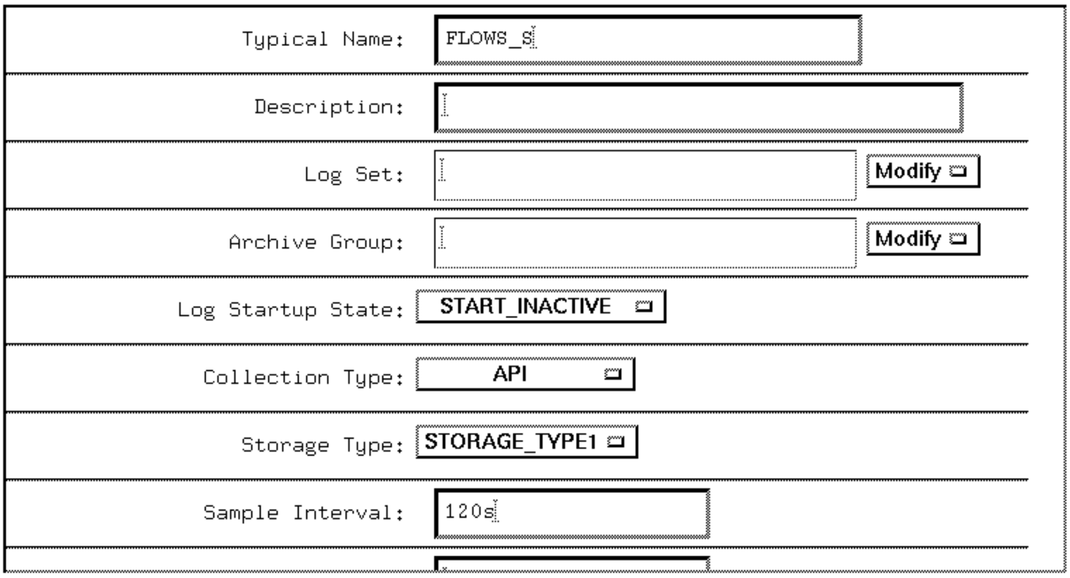
The window contains more attributes than can be displayed at one time. You can use the scroll-bar on the side of the display to scroll through the whole list of attributes.

4. To save the log typical when you are finished defining the attributes, choose **Save** from the **File** Menu.
5. This displays a dialog box with a list of History Managers.
6. Select a History Manager from the list, and then click on **OK**.

   This saves the log typical to the node where that History Manager resides.
7. Repeat these steps to build the second log typical.
   a. First clear the Log Typical window by choosing **New** from the **File** menu.

      This clears all the fields so that you can define a new log typical from scratch.
   b. Use the attribute values shown in Figure 3-36.

c. After saving, choose **Exit** from the **File** menu to close this window.



Typical Name: FLOWS_S

Description:

Log Set:

Archive Group:

Log Startup State: START_INACTIVE

Collection Type: API

Storage Type: STORAGE_TYPE1

Sample Interval: 120s

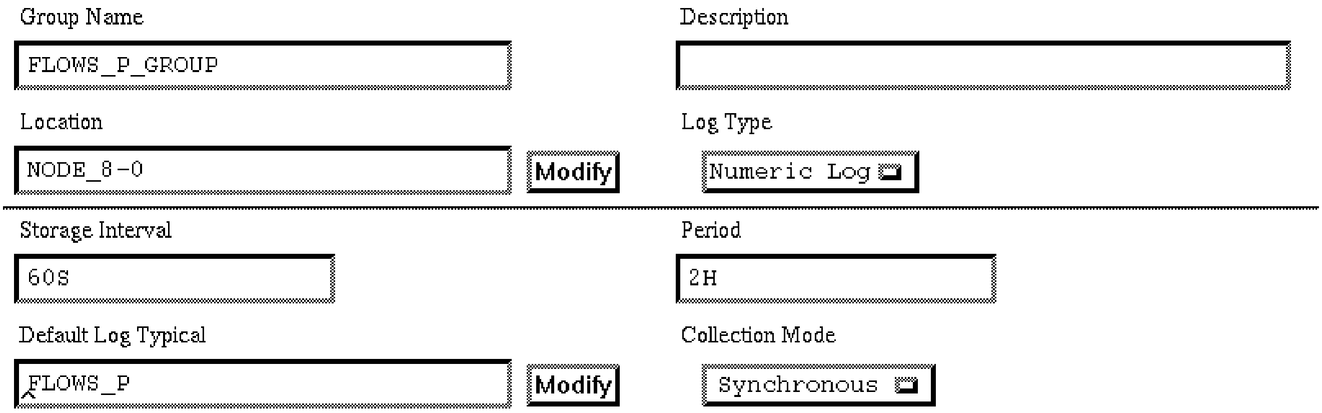
Figure 3-36. FLOWS_S Log Typical Definition

3.4.3.2 Build Log Groups

Build the log groups and assign the log typicals to them. Two log groups are required. One for primary logs and one for secondary logs. To build these log groups:

1. Choose **Log Group** from the **Tools** menu in the History Configuration window.

   This opens the Log Group window. Figure 3-37 shows the attribute values for the primary log group.



Group Name: FLOWS_P_GROUP

Description:

Location: NODE_8-0

Log Type: Numeric Log

Storage Interval: 60S

Period: 2H

Default Log Typical: FLOWS_P

Collection Mode: Synchronous

Figure 3-37. FLOWS_P_GROUP Definition

2. Enter **`FLOWS_P_Group`** in the Group Name field.
3. Leave Location at the default (`NODE_8-0`).
4. Leave the Log Type at its default (`NUMERIC`).
5. Enter **`60s`** (60 seconds) in the Storage Interval field.
6. Enter **`2H`** (2 hours) in the Period field.
7. Leave the collection field at the default (`SYNCHRONOUS`).
8. Enter a default log typical in the Default Log Typical field. To enter a default log typical, click on the **Modify** button by the Log Typical field. This displays a list of log typicals to choose from, Figure 3-38.

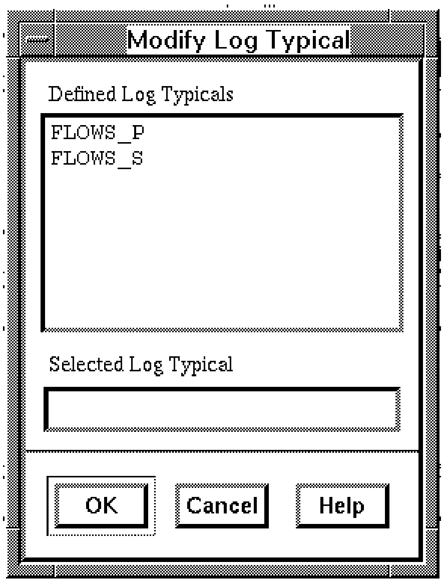


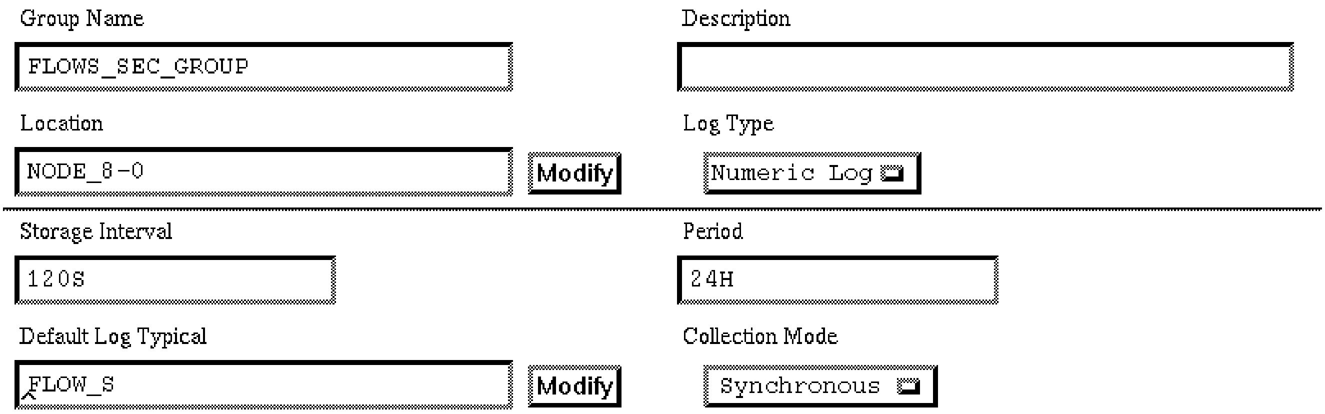
Figure 3-38. Modify Log Typical Dialog Box

9. Choose `FLOWS_P`, and then click on **OK**.

   This closes the dialog box. The selected default log typical is entered in the Default Log Typical field in Log Group window.

10. Choose **Save** from the **File** menu to save the log group.

11. Build a second log group for secondary logs as described above.
    a. First clear the log group window by choosing **New** from the **File** menu.
    b. Use the attribute values shown in Figure 3-39.
    c. After saving, choose **Exit** from the **File** menu to close the Log Group window.



Group Name: FLOWS_SEC_GROUP
Description:
Location: NODE_8-0
Log Type: Numeric Log
Storage Interval: 120S
Period: 24H
Default Log Typical: FLOW_S
Collection Mode: Synchronous

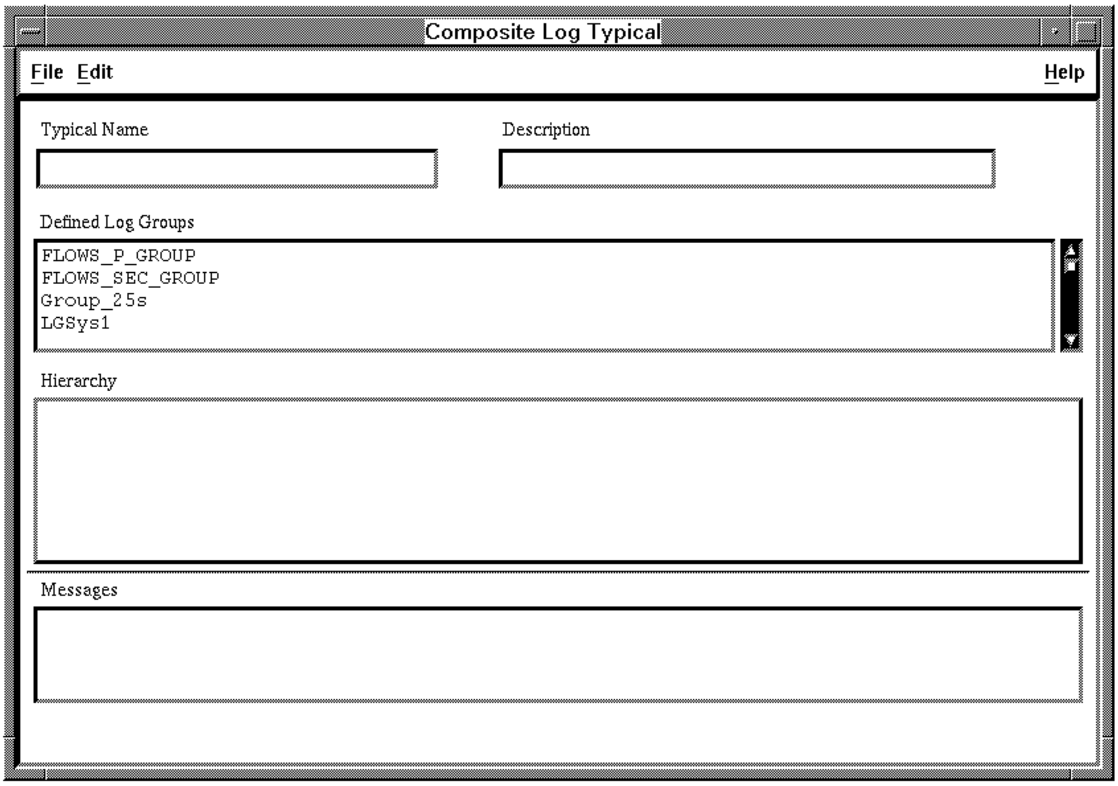
Figure 3-39. FLOWS_SEC_GROUP Definition

3.4.3.3 Build a Composite Log Typical

Build the Composite Log Typical to define the log hierarchy. To build a composite log typical:

1. Choose **Composite Log Typical** from the **Tools** menu in the History Configuration window.

This opens the Composite Log Typical window, Figure 3-40. Initially, just the available log groups are listed in the Log Groups scroll box.

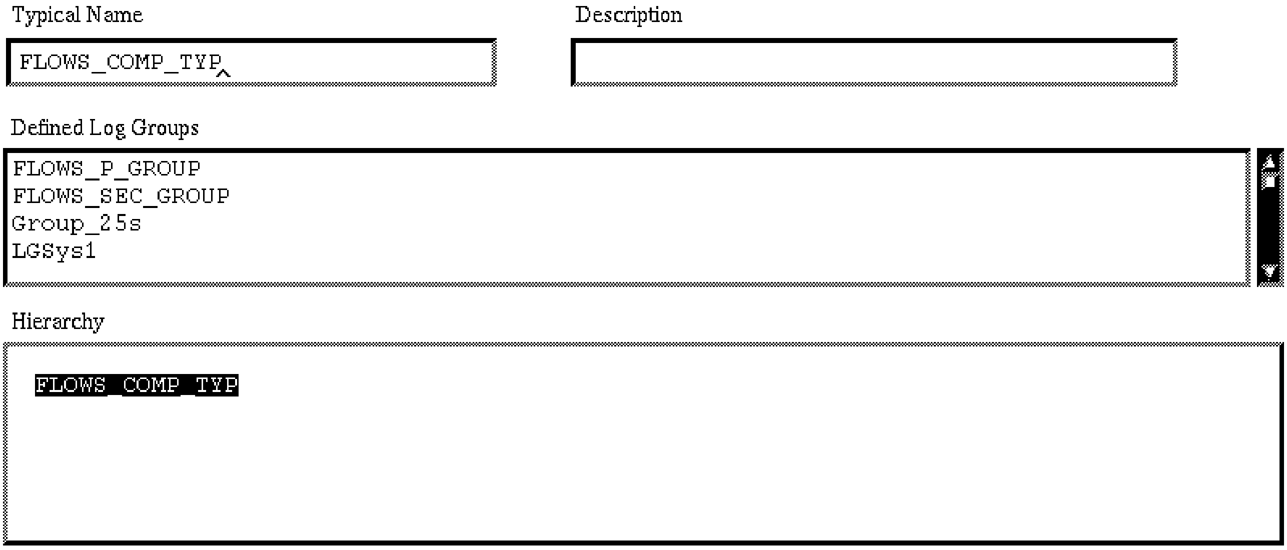


FLOWS_P_GROUP
FLOWS_SEC_GROUP
Group_25s
LGSys1

Figure 3-40. Composite Log Typical Window

2. Enter the name of the composite log typical in the Typical Name field. As you enter the name in the Typical Name Field, it is also entered in the Hierarchy Box.

3. Select the Typical Name in the hierarchy box, Figure 3-41.



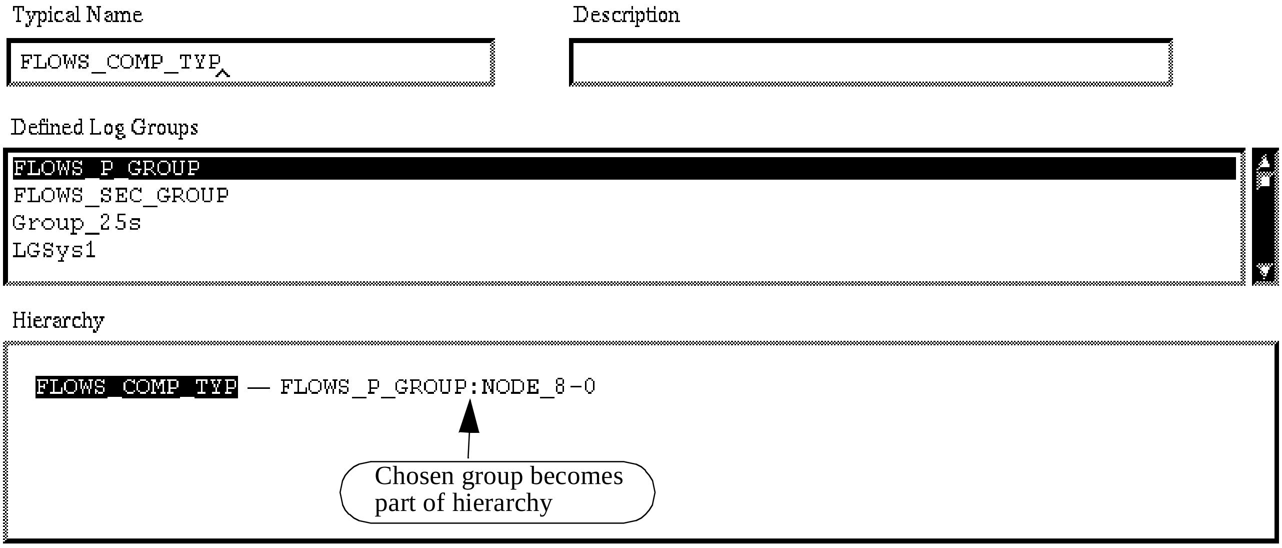
FLOWS_COMP_TYP

FLOWS_P_GROUP
FLOWS_SEC_GROUP
Group_25s
LGSys1

FLOWS_COMP_TYP

Figure 3-41. Typical Name Selected

4. Choose a primary group from the Log Groups Box, Figure 3-42.



FLOWS_COMP_TYP

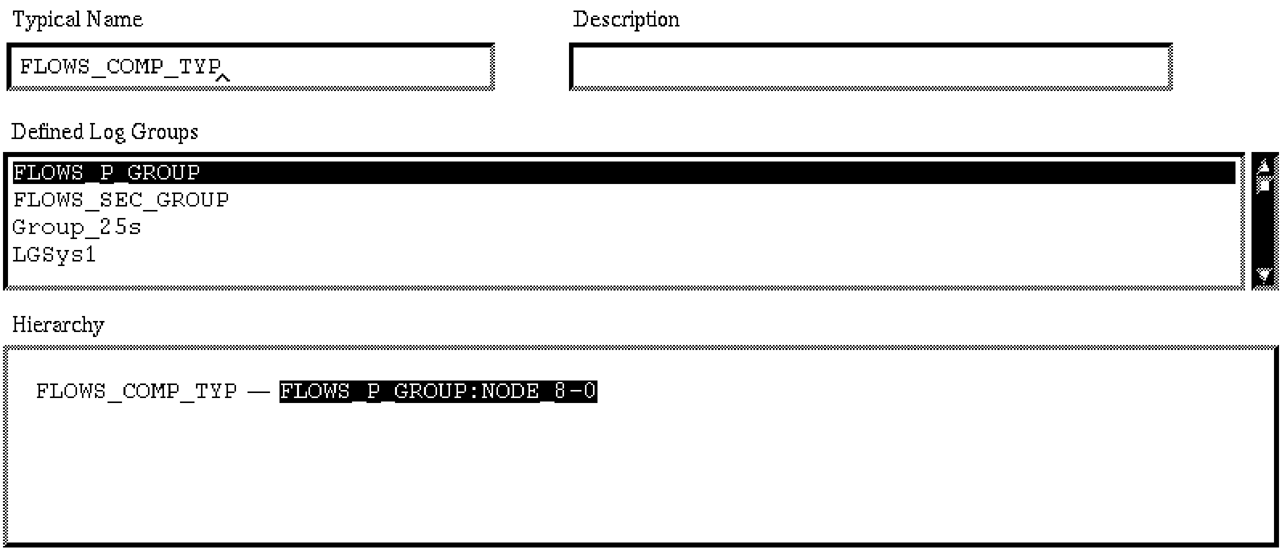
FLOWS_P_GROUP
FLOWS_SEC_GROUP
Group_25s
LGSys1

FLOWS_COMP_TYP — FLOWS_P_GROUP:NODE_8-0

Figure 3-42. Selecting the Primary Log Group

As soon as you select the primary log group, it becomes part of the log hierarchy.

5. Select the primary group in the Hierarchy box to specify where to add the secondary log. (Secondary log will be added to the right of the selected group), Figure 3-43.



Typical Name
FLOWS_COMP_TYP

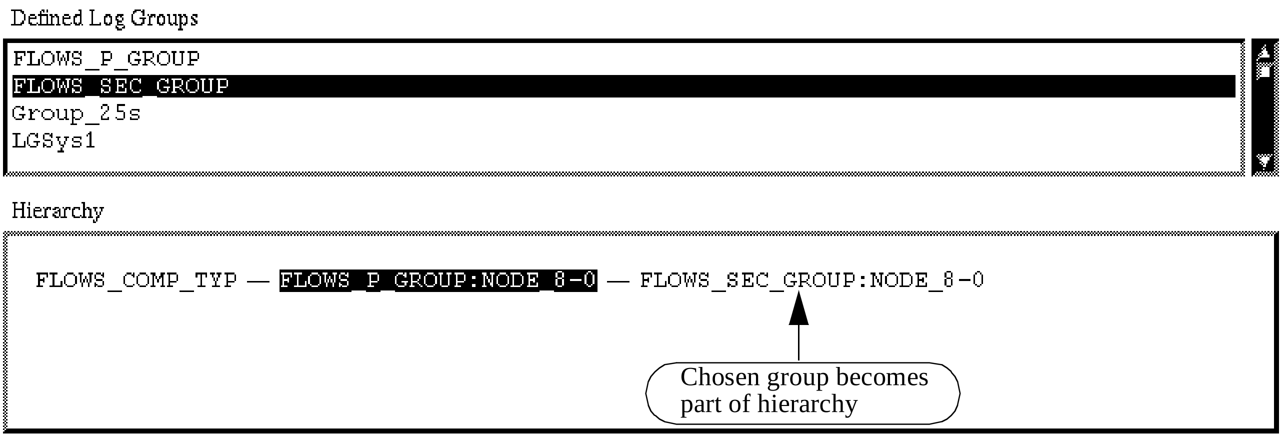
Description

Defined Log Groups
FLOWS_P_GROUP
FLOWS_SEC_GROUP
Group_25s
LGSys1

Hierarchy
FLOWS_COMP_TYP — FLOWS_P_GROUP:NODE_8-0

Figure 3-43. Selecting Where Secondary Log Group Will Be Added

6. Select a secondary log by choosing another log group from the Log Groups box, Figure 3-44.



Defined Log Groups
FLOWS_P_GROUP
FLOWS_SEC_GROUP
Group_25s
LGSys1

Hierarchy
FLOWS_COMP_TYP — FLOWS_P_GROUP:NODE_8-0 — FLOWS_SEC_GROUP:NODE_8-0

Figure 3-44. Selecting the Secondary Log Group

The secondary log is added to the hierarchy.

7. To save the composite log typical, choose **Save** from the **File** menu.

This displays a dialog box with list of History managers. Select a History Manager from the list as the location for the composite log typical and then click on **OK** to close the dialog box.

8. Choose **Exit** from the **File** menu to close the window.

Now you have a composite log typical to use as a template for new logs. Composite log typicals are applied to logs via the Composite Log window as described below.

3.4.3.4 Build Composite log using a Composite Log Typical

Build the composite logs with the Composite Log Window as follows:

1. Choose **Composite Log** from the **Tools** menu in the History Configuration Window.

   This opens the Composite Log window.

2. Specify the data source for the log in the Access Name field (for example `F100,MEASURE or F100,VALUE`.

3. Select a composite log typical to specify the log hierarchy.

   First click on the data source in the Hierarchy box. Then press and hold the right mouse button to display a pop-up menu. Drag down to **ADD COMP LOG TYP** and release to select this option. This displays the Add Composite Log Typical dialog box, Figure 3-45

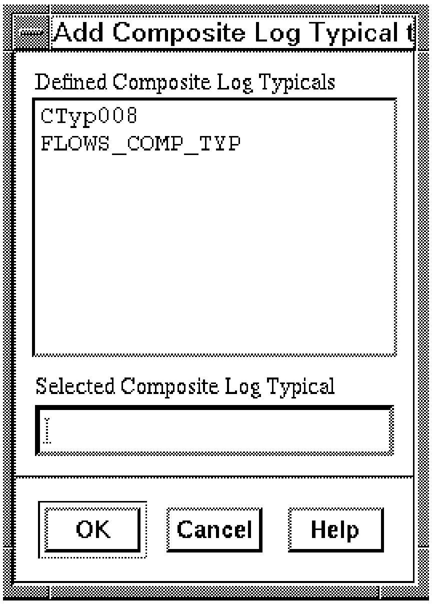
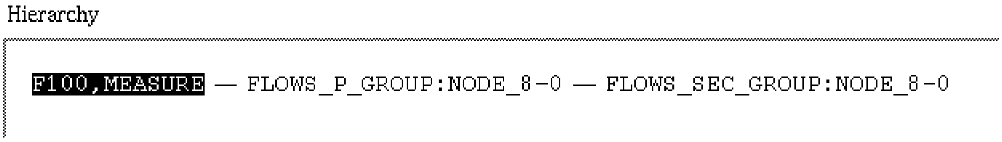

Add Composite Log Typical t
Defined Composite Log Typicals
CTyp008
FLOWS_COMP_TYP
Selected Composite Log Typical
OK
Cancel
Help

Figure 3-45. Selecting a Composite Log Typical

Choose `FLOWS_COMP_TYP` from the Defined Composite Log Typical box, then click on **OK**. This returns you to the Composite Log window.

The Composite Log window now displays the selected composite log typical hierarchy in its Log Hierarchy Box, Figure 3-46

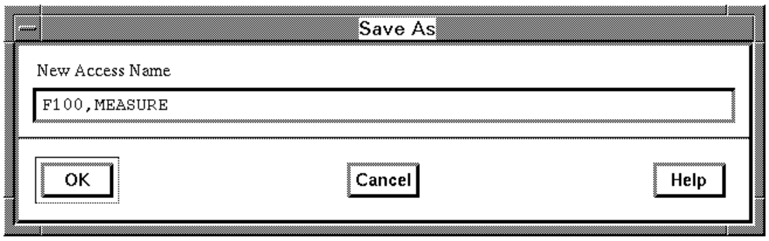


F100,MEASURE — FLOWS_P_GROUP:NODE_8-0 — FLOWS_SEC_GROUP:NODE_8-0

Figure 3-46. Composite Log Typical Hierarchy

4. To save this configuration choose **Save** from the **File** menu.
5. Build a second log using the **Save As** menu item of the **File** menu as follows:
   a. Choose **Save As** from the **File** menu.

      This displays the Save As dialog box, Figure 3-47.



New Access Name: F100,MEASURE

Figure 3-47. Save As Dialog Box

   b. Enter a new data source for the second log (for example F101,MEASURE or F101,VALUE) and then click **OK**. This produces a new log which is a copy of the original log with a new access name and data source.

3.5 Application Procedures

Generally, you use a combination of two or more of these application procedures to configure a History application. This section provides guidelines for using the application procedures to achieve a specific History application.

3.5.1 Getting Started

All History building applications begin with opening the History configuration window and selecting a History Manager. For details refer to Section 3.6.3, History Configuration Window.

3.5.2 Create History Directories

If you use file-based storage (Storage Type = TYPE1 or TYPE2), you must specify where to store the file-based logs. This involves specifying the directory (or disk), and how much of the disk is to be allocated to History. Allocation of disk space must be done before any file-based logs can be created. You can perform this function on-line via the Directory Maintenance window as described in Section 3.6.11, Directory Maintenance Window, or off-line using the hsDBMaint functions as described in Section 5.6.8, Add, Alter, Delete, or List Numeric Log Directories.

3.5.3 Log Sets

Log sets should be configured before you build the logs to be assigned to them. This is done via the Log Set window as described in Section 3.6.7, Log Set Window. You can assign logs to log sets via the Composite Log or Log Typical window as described in Section 3.5.3.1, How to Assign Logs to a Log Set.

3.5.3.1 How to Assign Logs to a Log Set

There are two ways to start and stop related groups of logs as one unit. One way is via the log group. This way you can start or stop all the logs in the same log group. This is somewhat limiting because all the logs must be of the same type, have the same storage interval, and other attributes defined by the log group. Another method for grouping related logs is via the log set. In a log set, you can include logs of different types, having different storage intervals and so on. You assign logs to a log set via the Composite Log window. Typically, this is done while you are configuring the log in the Composite Log window. How to use the Composite Log window to assign a log to a log set is described below.

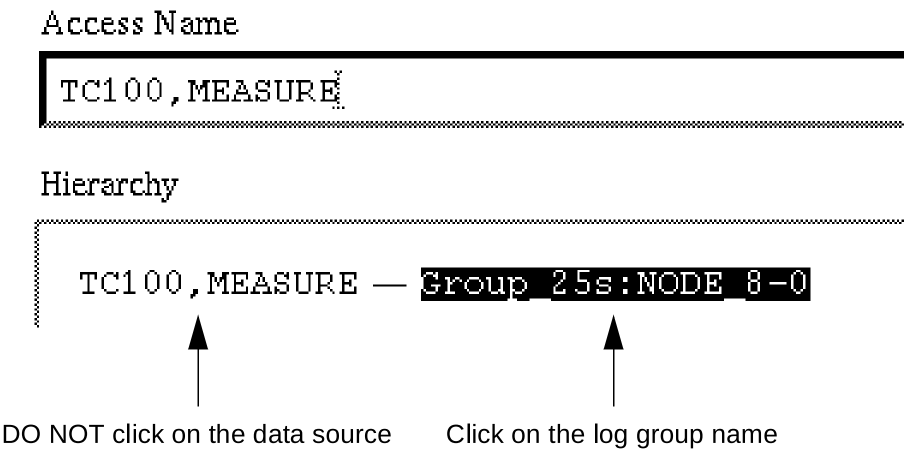
To assign logs to a log set:

1. Open the Composite Log window (if not already opened).
2. Choose **Open** from the **File** menu.

   This displays a list of existing composite logs.
3. Select the log you want to open, then click on **OK**.

   This opens the log into the Log Composite window.
4. Click on the log in the Hierarchy box, Figure 3-28.

Click on the log group name and NOT the data source to open the edit window for a log. This is because it is possible to have multiple logs connected to the same data source (via a composite log hierarchy). This is the only means to distinguish one log from another.



Access Name

TC100,MEASURE

Hierarchy

TC100,MEASURE — Group_25s:NODE_8-0

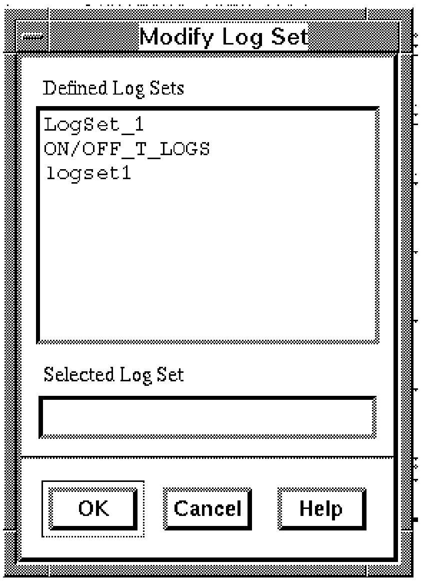
DO NOT click on the data source

Click on the log group name

Figure 3-48. Selecting Log to Open

5. Choose **Modify Log Set** from the **Log** menu.

   This displays a Modify Log Set section box, Figure 3-49.



Defined Log Sets

LogSet_1
ON/OFF_T_LOGS
logset1

Selected Log Set

Figure 3-49. Modify log Set Dialog Box

6. Choose the log set that you want to assign the current log to, then click **OK**. Remember that all logs in a log set must be in the same node as the log set.

3.5.4 A Simple Log without using Typicals

To configure a basic log:

1. Configure a log group via the Log Group window, Section 3.6.6, Log Group Window.
2. Configure logs for the group via the Composite Log window, Section 3.6.8, Composite Log Window.

3.5.5 Simple Logs using a Log Typical

To configure logs that are very similar (or identical, except for data source):

1. Configure a log typical, Section 3.6.5, Log Typical Window.
2. Configure a log group, Section 3.6.6, Log Group Window. Use the log typical you configured in step 1 as the default log typical for the log group.
3. Configure logs for the group, Section 3.6.8, Composite Log Window.

3.5.6 Composite Logs using a Log Typical and Composite Log Typical

To configure multiple composite logs that are very similar (or identical, except for data source):

1. Configure a log typical, Section 3.6.5, Log Typical Window.
2. Configure a log group, Section 3.6.6, Log Group Window. Use the log typical you configured in step 1 as the default log typical for the log group.
3. Configure a composite log typical, Section 3.6.4, Composite Log Typical Window.
4. Configure logs, Section 3.6.8, Composite Log Window.

3.5.7 Configure History Node to Support Archive Function

To configure your History node to support archive functions:

1. Configure the archive device. Use the Archive Device window to assign a name to the DAT drive, specify the History node that the DAT drive is connected to, assign a name to the device file, and activate the device. Refer to Section 3.6.9, Archive Device Window.
2. Configure archive groups. Refer to Section 3.6.10, Archive Group Window.
3. Assign logs to archive groups. This is done via the Composite Log window when you configure composite logs, Section 3.6.8, Composite Log Window.

3.5.8 How to Build Report Logs

Report logs are simple asynchronous primary logs. You have flexibility in determining how to apply them to your application. Report applications usually dedicate one report log to one report, that is, if you have a WKLY_SUM report, you make a log especially for it. You put each week's version of the WKLY_SUM report into the report log and use different report logs for other reports. However, you are not required to do it this way. You are free to send more than one type of report to one report log.

You can save finished reports from the AdvaInform Reports package and encapsulated data from the reports in AdvaInform History. To save reports, all you have to do with the History package is build the report logs to store the reports. In AdvaInform Reports, when you create a report object instance or execute a report manually, you must enable the History execution parameter and enter the name of the report log as the History execution parameter value.

You build report logs in the Composite Log window in the same manner that you build other log types. The report log attributes are described in Table 3-5.

When you build a report log, keep the following in mind:

- A report log does not have to be activated in order to store reports. Therefore, log sets are not applicable for report logs.
- The log capacity for a report log defines the number of reports it can hold. You must specify a log capacity for report logs.
- The data collection mode for report logs is asynchronous. This is configured in the log group for the report log. Storage interval and log period are not applicable, and therefore, do not need to be configured in the log group.
- The access name that you configure in the composite log window is the name you use to identify the log when you send reports to it.

The basic steps are as follows:

1. Build a report log group with asynchronous collection mode. Do not enter values for Storage Interval and Period for the Log Group. You can use a log typical. If you choose not to use one, the report log is generated with default attribute values. The Log Group must be built before you build the report log.
2. Build the report log in the Composite Log window.
3. Open the Log Attribute Edit Window for the log and enter a value for the Log Capacity attribute.

3.5.9 How to Build Message Logs for MOD 300

3.5.9.1 Advant OCS Database Requirements for History Message Logs

History can store messages from CCF and TCL. The messages must be routed to History. To support this functionality you must configure the applicable database objects (templates) in the Advant OCS database. Refer to the *Advant Station 500 Series IMS User's Guide* for details.

The objects (templates) that are related to routing messages to History are:

History	You build one History object to specify that History is to execute in your History node (Advant Station). This object is a child of the GENERICD (Generic DPSS) object for the History node.
Recorder	This object does not require any specific entries. It merely has to be present in the Advant OCS database. The Recorder object is a child of the History object. In addition to supporting message routing, this object also supports the ability to view AdvaInform History data on a Multibus-based trend display.
Configuration Area	You use the message routing edit window to route CCF and TCL messages that originate in the configuration area to History. You have a choice of routing any combination of CCF alarm, CCF event, TCL event, and TCL unit (billboard) messages. You identify History in the History node by the name you assigned to the History Template. For example to route all CCF and TCL messages from a configuration area to a History package named HIS_IMS, the edit window entries are

MESSAGE CTR	MESSAGE TYPE	REMOTE TYPE
HIS_IMS	CCF_BOTH	
HIS_IMS	TCL_BOTH	

Message Routing	Each Advant OCS Multibus-based console uses this object to route console log on and alarm acknowledgment messages. The routing is defined via entries to an edit window. You identify History in the History node by the name you assigned to the History object. For example to send both log on and alarm acknowledgment messages to History package HIS_IMS, the entries are:

MESSAGE CTR	MESSAGE TYPE	REMOTE TYPE
HIS_IMS	CONS_ACKS	
HIS_IMS	CONS_LOG_ONS	

3.5.9.2 Guidelines for Configuring Message Logs for MOD 300

You need to build only one message log per History node to store CCF, TCL, and console messages. A message log contains the log attributes that are common to all history logs. They are described in Table 3-5.

You build message logs in the Composite Log window in the same manner that you build other log types. When you build message logs, keep in mind that:

- To store CCF and TCL messages, build a message log with the name: MODMSGLOG_x_y (where *x* and *y* are the device and subdevice addresses of the node). Event messages from AdvaInform Basic (User) objects are stored in a message log with the name: EAMSGLOG. This log is created when History software is installed. Other message logs can be built to store messages from other sources.
- The message log must be activated in order to store messages. This can be done via the log group or log set. A log group must be configured for a message log. A log set is optional.
- The log capacity for a message log defines the number of messages it can hold. You must specify a log capacity for the message log.
- The data collection mode for a message log is asynchronous. This is configured in the log group for the message log. Storage interval and log period are not applicable, and therefore, do not need to be configured in the log group.

The basic steps are as follows:

1. Build a message log group with asynchronous collection mode. Do not enter values for Storage Interval and Period for the Log Group. You can use a log typical. If you choose not to use one, the message log is generated with default attribute values. The Log Group must be built before you build the message log.
2. Build the message log in the Composite Log window.

   For CCF and TCL messages, enter the access name for the message log as follows:

   `MODMSGLOG`_*x*_*y* (for example: **MODMSGLOG_8_0**)
3. Open the Log Attribute Edit Window for the log and enter a value for the Log Capacity attribute.

3.5.10 Configure Logs to be Archived with Log Typicals

To configure logs to be archived with log typicals:

1. Configure the archive device, Section 3.6.9, Archive Device Window.
2. Configure an archive group to reference the archive device, Section 3.6.10, Archive Group Window.
3. Configure a log typical to reference the archive group, Section 3.6.5, Log Typical Window.
4. Configure a log group with reference to the log typical, Section 3.6.6, Log Group Window.
5. Configure a composite log from the log typical, Section 3.6.8, Composite Log Window.

3.5.11 How to Edit Log Attributes

Limitations

When you configure a log in the Composite Log Window, before you save the log you can edit most log attributes. Attributes that you can not edit are:

- Log Group
- Collection Mode
- Storage Interval
- Number of Samples for Collection
- Log Capacity (for synchronous logs)

After you save the log, the following attributes can not be edited (in addition to the ones above):

- Access Name
- Object Name
- Calculation Algorithm

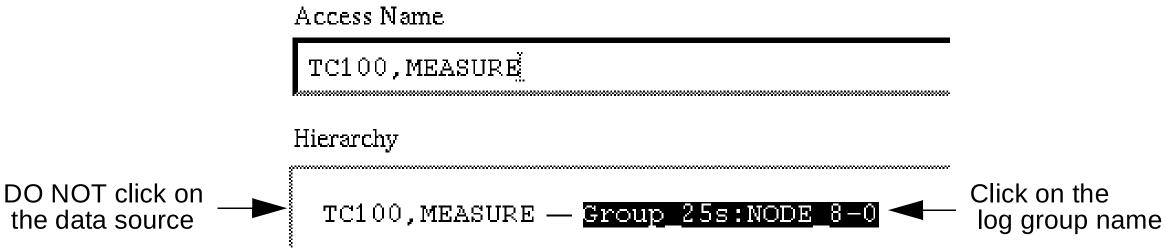
It is recommended that the presentation, engineering units, and alarm attributes as described in Section 3.1.5, Presentation (Trend), Engineering Units, and Alarm Attributes not be edited.

Procedure

To edit log attributes:

1. Open the Composite Log window. Initially the window is empty.
2. Choose **Open** from the **File** menu. This displays a list of existing logs.
3. Select the composite log you want to edit, then click on **OK**. This opens the log into the Log Composite window.
4. Click on the log in the Hierarchy box, Figure 3-50.

   Click on the log group name and NOT the data source to open the edit window for a log. This is because it is possible to have multiple logs connected to the same data source (via a composite log hierarchy). This is the only means to distinguish one log from another.



Access Name

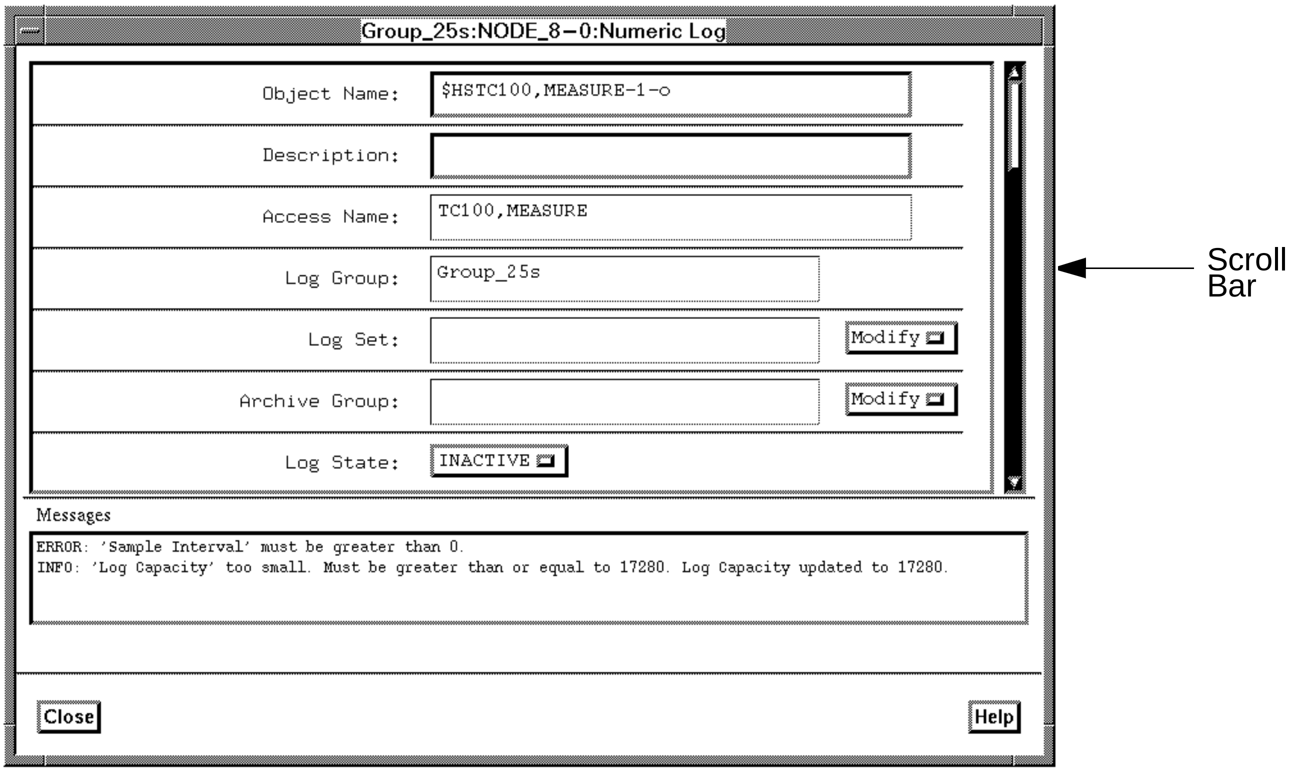
TC100,MEASURE

Hierarchy

TC100,MEASURE — Group_25s:NODE_8-0

Figure 3-50. Selecting a Log to Edit

5. Choose **Open** from the **Log** menu. This displays the log attribute edit window for the selected log, Figure 3-51.



Group_25s:NODE_8–0:Numeric Log

Object Name: $HSTC100,MEASURE-1-o
Description:
Access Name: TC100,MEASURE
Log Group: Group_25s
Log Set:
Archive Group:
Log State: INACTIVE

Messages

```
ERROR: 'Sample Interval' must be greater than 0.
INFO: 'Log Capacity' too small. Must be greater than or equal to 17280. Log Capacity updated to 17280.
```

Figure 3-51. Log Attribute Edit Window

6. Check that the log state is INACTIVE. If it is ACTIVE, deactivate the log. You are not allowed to change attributes in an ACTIVE log.
7. Make any modifications required to the log's attributes. Use the scroll bar if it is necessary to scroll through the attribute list.
8. Click on **Close** when you are finished editing the attributes. This returns you to the Composite Log window.
9. To save the changes you have made, choose **Save** from the **File** menu in the Composite Log window. The modified attributes are not saved in the log until this step is performed.

3.5.12 Error Messages

When a procedure fails, a message is displayed in the Message box of the affected log. For example:

```
ERROR: Dependency Check Failed. See Log Message for
$HSFC100,VALUE-1-o.
```

This indicates that you should open the primary log and read its Message box. Logs are identified in the following manner in the messages, Figure 3-52.

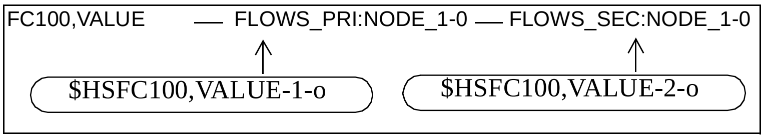


Figure 3-52. Logs in Messages

3.5.13 How to Configure History Logs in a Text File

Typically, the bulk of the History database is made up of synchronous numeric logs. For such applications, AdvaInform History provides two utility programs (dbDump and dbLoad) that facilitate the configuration of large numbers of numeric logs. dbDump creates an ASCII text file from a history database template that you configure with the History Builder. This template must include all History objects, except the majority of the numeric objects. You must include one numeric object for each numeric log group. These logs are used as templates to create the bulk of the numeric logs in the database. The additional tags that require logs are specified in tag lists that you create using a text editor. dbDump reads the tag lists and creates another text file with “like” statements that specify how to configure logs for the listed tags. dbLoad configures the complete History database from the text files created with dbDump. dbLoad checks the syntax of the loaded files and verifies the configuration information. If there are no errors, dbLoad creates the history objects defined in the files for the applicable node. This method for defining logs is quicker than the Save As method.

For History on an Advant Station with AdvaInform software, it is recommended that you be logged into the IMS menu as an ocsmgr user to use these utility programs. If you are using the History menu on an Advant Station with AdvaCommand software, you should be logged in as Operator.

The procedure for configuring a History database with text files is described in Section 3.5.13.1, Procedure for Configuring History Database with Text Files. A second procedure for dual history is provided in Section 3.5.13.2, Dual History Example (MOD 300 Only).

Details regarding the operation of the dbDump and dbLoad utilities are provided in Section 3.5.13.3, dbDump and dbLoad Usage.

NOTE

Do not start by creating text files. Create your History objects via the History Builder first. The following sequence is recommended:

1. Use History Builder to create History objects such as log groups, log sets, report logs, message logs and so on.
2. Use History Builder to create the template numeric logs.
3. Use text editor to create tag lists (one file per numeric log template) for additional numeric logs.
4. Use dbDump to read History database (configured via History Builder) into a text file.
5. Use dbDump to create a log_like file for numeric log template.
6. Use dbLoad to parse and check all text files created via dbDump.
7. Drop/recreate the database (this is optional).
8. Use dbLoad to configure the History Database from the Text Files
9. Use hsBAR to create a backup copy of your database.

3.5.13.1 Procedure for Configuring History Database with Text Files

The following procedure is based on an example application that requires History to collect from 1000 tags in controllers with the following rates and periods:

- 100 logs: sample rate 6s, storage rate 6s, log period 3d, deadband of 5:1 expected.
- 400 logs: sample rate 15s, storage rate 15s, log period 3w, deadband not configured.
- 500 logs: sample rate 30s, storage rate 30s, log period 3w, deadband not configured.

This example case requires a minimum of 3 composite logs to be configured in the database template via the History Builder; LOG1 for the 100 logs, LOG2 for the 400 logs, and LOG3 for the 500 logs.

LOG1 requires a 6s log group. If archive is required, an archive time interval of 1 day would be appropriate. (The log is guaranteed to have 3 days of data, a time interval of 1 day would provide more chances to archive data if an archive fails.) A time interval of 3 days would not guarantee archival of all data if deadband was configured incorrectly and a point was stored every 6 seconds.

LOG2 requires a 15s log group. If archive is required, an archive time interval of 3d would be appropriate. (The log is guaranteed to have 3d of data, a time interval of 3d would provide 6 more chances to archive data if an archive fails.)

LOG3 requires a 30s log group. If archive is required, an archive time interval of 3d would be appropriate. (The log is guaranteed to have 3d of data, a time interval of 3d would provide 6 more chances to archive data if an archive fails.)

NOTE

This example is for an application where History is operating on a single node. Following this example, a second example is provided to demonstrate how to implement the same application with dual History.

The files produced for this example application (for both standard and Dual History) are provided in the following directory:

```
/opt/advant/History/Examples
```

You can refer to these files for reference. To open them, use a text editor such as vi.

Use the History Builder to Configure a Database Template

To build the database template:

1. If you intend to do any archiving, create an Archive Device.
2. If REPORT logs are needed, create the necessary Log Groups(s) and Report Log(s). Include Report Logs in Archive Groups as desired.

   Choose the appropriate log capacity for the Report Log and determine the best archive time for timed archive. For example, consider a Report Log that holds 48 reports and a report is generated every2 hours. The TIMED archive time period should archive 1/4 to 1/3 of the reports every time the archive runs. For the best performance for this example, use an archive time interval of 24 to 32 hours. For ease of scheduling, 1 day works well.

3. If message logs are needed, create the necessary Log Group(s) and Message Log(s).

   Create Message Archive Groups as needed. The same rules used for report logs also apply to message logs. If you create a message log with a capacity of 10,000 and expect 2000 messages per day, the message log will have approximately 5 days of messages. Therefore, an archive group time interval of one or two days is reasonable. The one-day interval should be sufficient.

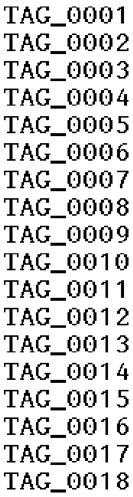
4. Build a log group and one composite log for each of the numeric log types (LOG1, LOG2, and LOG3) described above.

Use a Text Editor to Create Tag Lists for Additional Logs

Use a text editor such as vi to create tag list text files for the additional numeric logs. Create a separate text file for each composite log you defined in the History database template. For this example application, you would create a text file tag list for the following categories:

- 100 logs, sample rate 6s, storage rate 6s, log period 3d, deadband of 5:1 expected.
- 400 logs, sample rate 15s, storage rate 15s, log period 3w, deadband not configured.
- 500 logs, sample rate 30s, storage rate 30s, log period 3w, deadband not configured.

Enter one tag per line in the tag list file. For this example, the text files are named 100.tags, 400.tags, and 500.tags respectively. An example of a tag list text file is shown in Figure 3-53.



```
TAG_0001
TAG_0002
TAG_0003
TAG_0004
TAG_0005
TAG_0006
TAG_0007
TAG_0008
TAG_0009
TAG_0010
TAG_0011
TAG_0012
TAG_0013
TAG_0014
TAG_0015
TAG_0016
TAG_0017
TAG_0018
```

Figure 3-53. Example Tag List Text File

Read History Database Template into a Text File

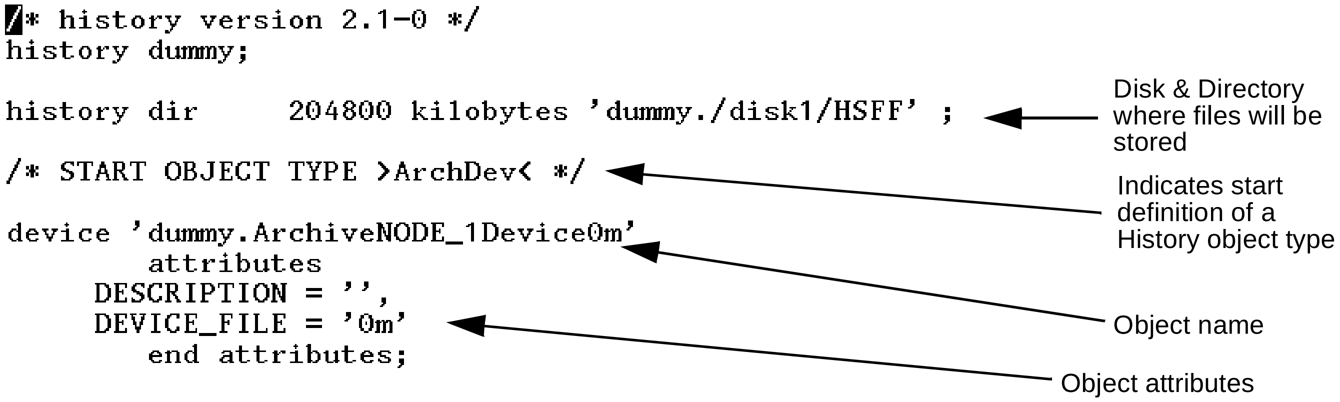
Use dbDump with the -d option to dump the template History database to a text file. For example:

```
dbDump -d > /tmp/template.db
```

In this case, the database is being dumped to a text file named template.db in the /tmp directory.

Once you have created a text file using dbDump, you can open the file to view and edit using your text editor. For example, if you use the vi text editor, enter the following command to open the text file dumpfile in the tmp directory: `vi /tmp/dumpfile`

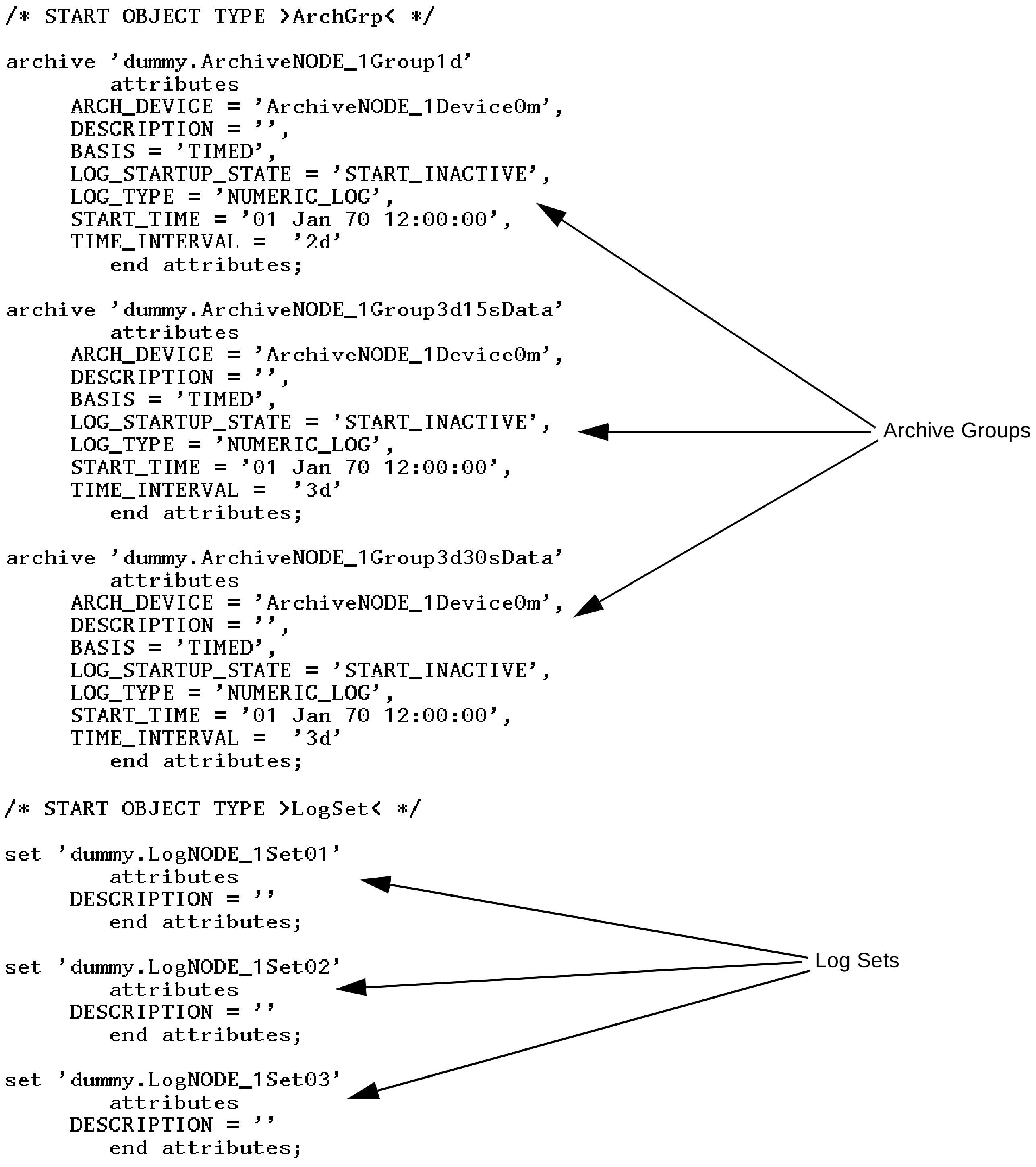
An example of the text file produced by this command is shown in Figure 3-54 to Figure 3-59.



```
/* history version 2.1-0 */
history dummy;

history dir     204800 kilobytes 'dummy./disk1/HSFF' ;

/* START OBJECT TYPE >ArchDev< */

device 'dummy.ArchiveNODE_1Device0m'
        attributes
     DESCRIPTION = '',
     DEVICE_FILE = '0m'
        end attributes;
```

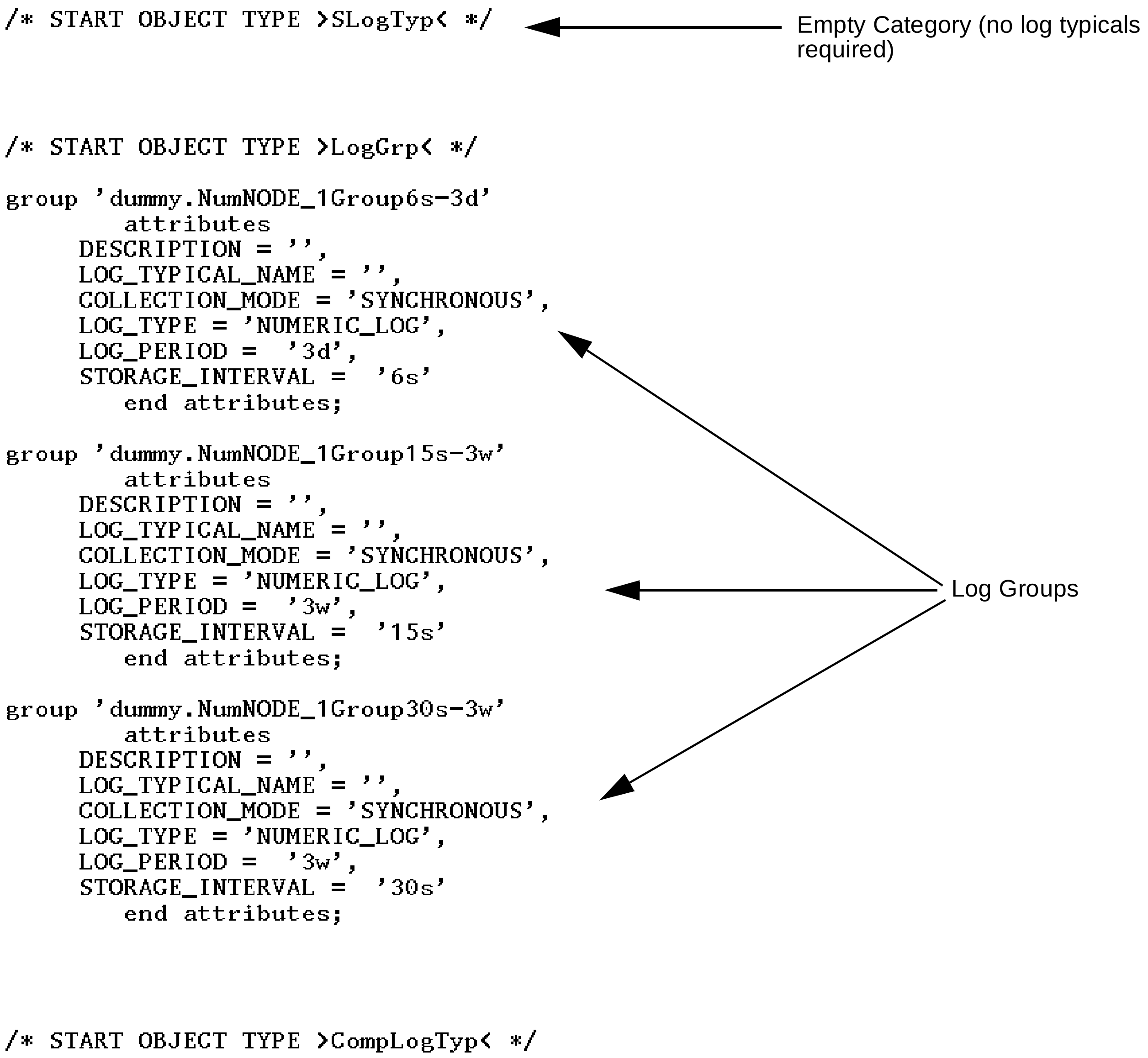
Disk & Directory where files will be stored
Indicates start definition of a History object type
Object name
Object attributes

Figure 3-54. Example, First Part of dbDump Text File



```
/* START OBJECT TYPE >ArchGrp< */

archive 'dummy.ArchiveNODE_1Group1d'
        attributes
     ARCH_DEVICE = 'ArchiveNODE_1Device0m',
     DESCRIPTION = '',
     BASIS = 'TIMED',
     LOG_STARTUP_STATE = 'START_INACTIVE',
     LOG_TYPE = 'NUMERIC_LOG',
     START_TIME = '01 Jan 70 12:00:00',
     TIME_INTERVAL =  '2d'
        end attributes;

archive 'dummy.ArchiveNODE_1Group3d15sData'
        attributes
     ARCH_DEVICE = 'ArchiveNODE_1Device0m',
     DESCRIPTION = '',
     BASIS = 'TIMED',
     LOG_STARTUP_STATE = 'START_INACTIVE',
     LOG_TYPE = 'NUMERIC_LOG',
     START_TIME = '01 Jan 70 12:00:00',
     TIME_INTERVAL =  '3d'
        end attributes;

archive 'dummy.ArchiveNODE_1Group3d30sData'
        attributes
     ARCH_DEVICE = 'ArchiveNODE_1Device0m',
     DESCRIPTION = '',
     BASIS = 'TIMED',
     LOG_STARTUP_STATE = 'START_INACTIVE',
     LOG_TYPE = 'NUMERIC_LOG',
     START_TIME = '01 Jan 70 12:00:00',
     TIME_INTERVAL =  '3d'
        end attributes;

/* START OBJECT TYPE >LogSet< */

set 'dummy.LogNODE_1Set01'
        attributes
     DESCRIPTION = ''
        end attributes;

set 'dummy.LogNODE_1Set02'
        attributes
     DESCRIPTION = ''
        end attributes;

set 'dummy.LogNODE_1Set03'
        attributes
     DESCRIPTION = ''
        end attributes;
```

Figure 3-55. Example, dbDump Text File, Archive Group & Log Sets

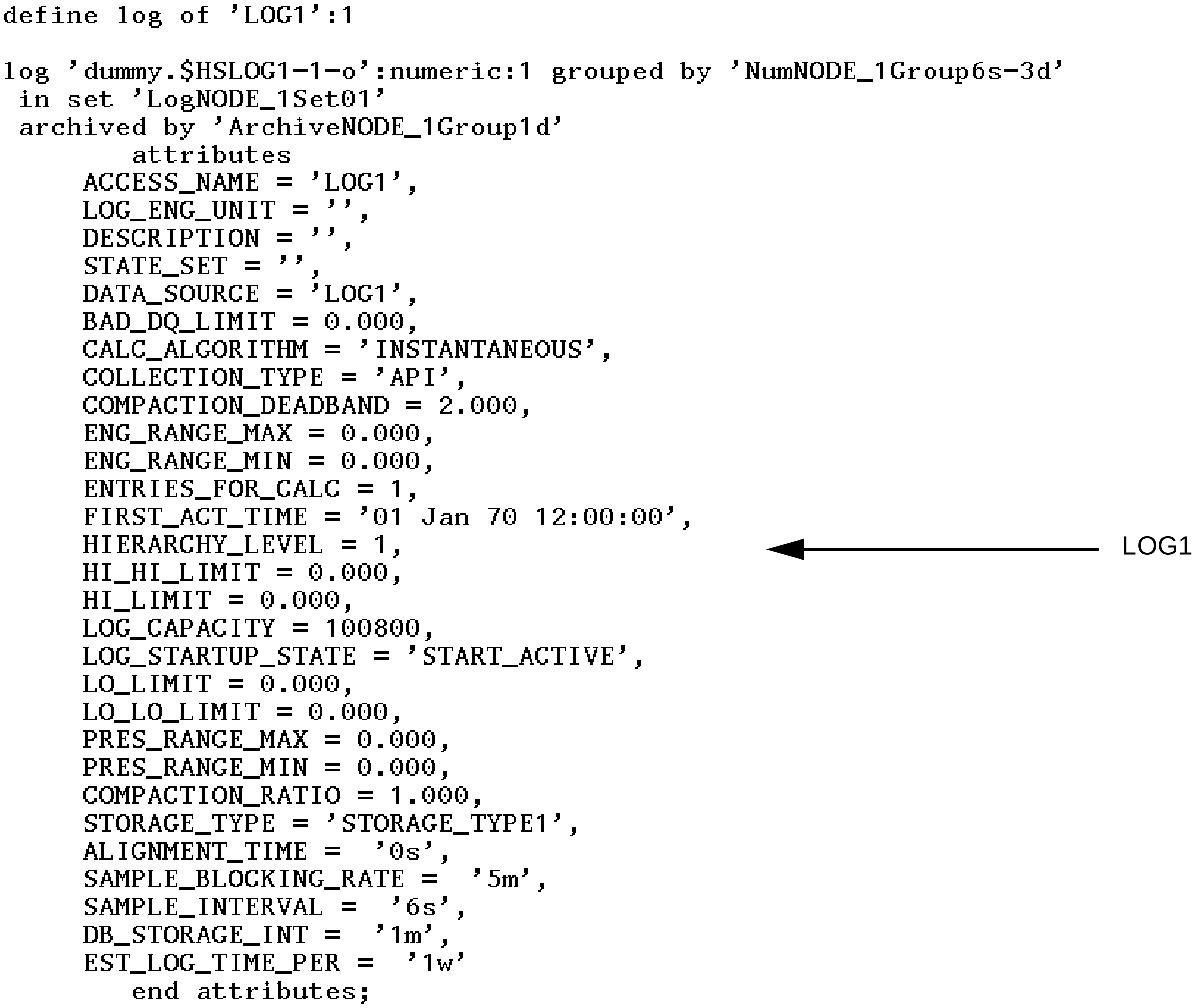


```
/* START OBJECT TYPE >SLogTyp< */


/* START OBJECT TYPE >LogGrp< */

group 'dummy.NumNODE_1Group6s-3d'
        attributes
     DESCRIPTION = '',
     LOG_TYPICAL_NAME = '',
     COLLECTION_MODE = 'SYNCHRONOUS',
     LOG_TYPE = 'NUMERIC_LOG',
     LOG_PERIOD =  '3d',
     STORAGE_INTERVAL =  '6s'
        end attributes;

group 'dummy.NumNODE_1Group15s-3w'
        attributes
     DESCRIPTION = '',
     LOG_TYPICAL_NAME = '',
     COLLECTION_MODE = 'SYNCHRONOUS',
     LOG_TYPE = 'NUMERIC_LOG',
     LOG_PERIOD =  '3w',
     STORAGE_INTERVAL =  '15s'
        end attributes;

group 'dummy.NumNODE_1Group30s-3w'
        attributes
     DESCRIPTION = '',
     LOG_TYPICAL_NAME = '',
     COLLECTION_MODE = 'SYNCHRONOUS',
     LOG_TYPE = 'NUMERIC_LOG',
     LOG_PERIOD =  '3w',
     STORAGE_INTERVAL =  '30s'
        end attributes;



/* START OBJECT TYPE >CompLogTyp< */
```

Empty Category (no log typicals required)

Log Groups

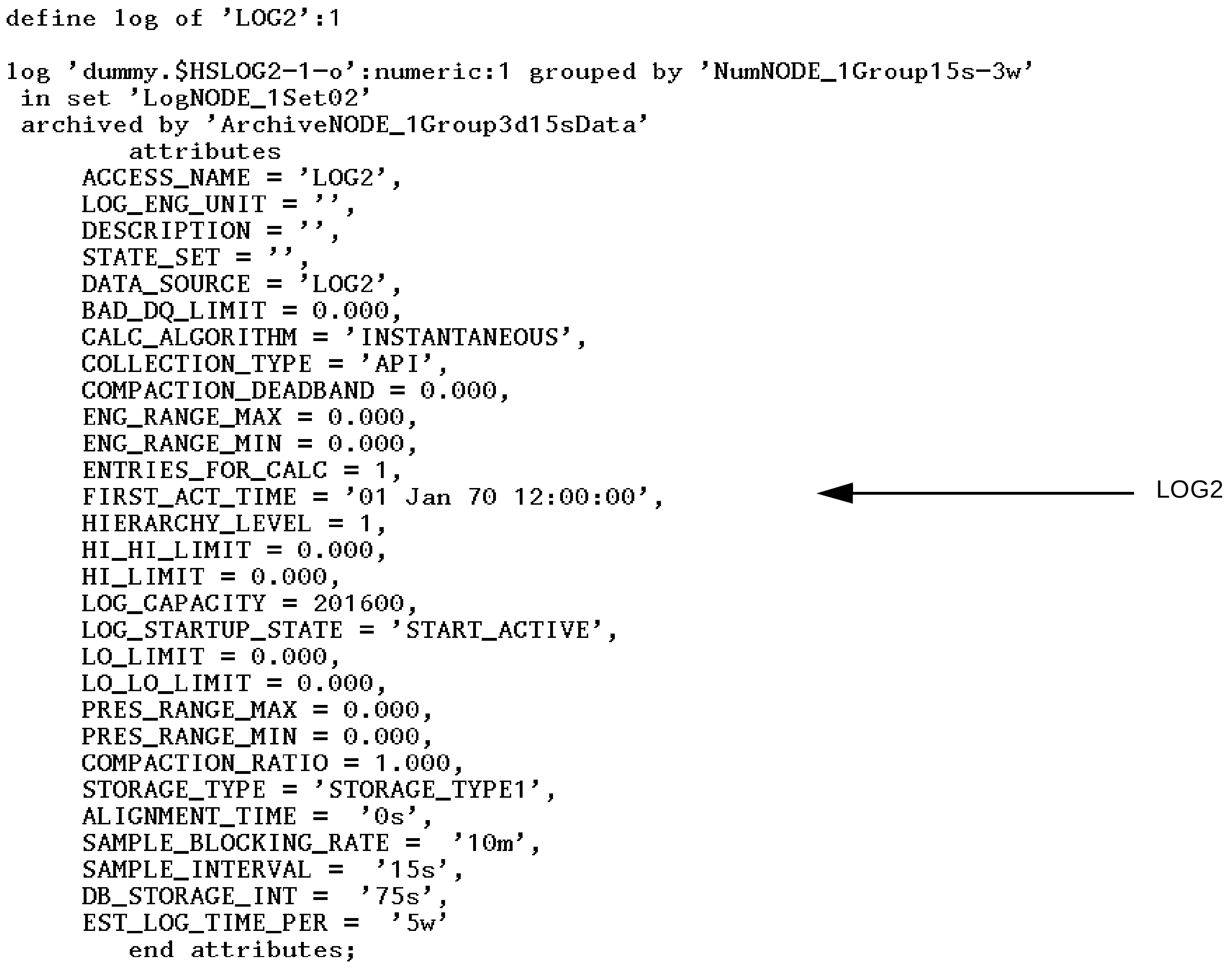
Figure 3-56. Example, dbDump Text File, Log Groups



```
define log of 'LOG1':1

log 'dummy.$HSLOG1-1-o':numeric:1 grouped by 'NumNODE_1Group6s-3d'
 in set 'LogNODE_1Set01'
 archived by 'ArchiveNODE_1Group1d'
        attributes
     ACCESS_NAME = 'LOG1',
     LOG_ENG_UNIT = '',
     DESCRIPTION = '',
     STATE_SET = '',
     DATA_SOURCE = 'LOG1',
     BAD_DQ_LIMIT = 0.000,
     CALC_ALGORITHM = 'INSTANTANEOUS',
     COLLECTION_TYPE = 'API',
     COMPACTION_DEADBAND = 2.000,
     ENG_RANGE_MAX = 0.000,
     ENG_RANGE_MIN = 0.000,
     ENTRIES_FOR_CALC = 1,
     FIRST_ACT_TIME = '01 Jan 70 12:00:00',
     HIERARCHY_LEVEL = 1,
     HI_HI_LIMIT = 0.000,
     HI_LIMIT = 0.000,
     LOG_CAPACITY = 100800,
     LOG_STARTUP_STATE = 'START_ACTIVE',
     LO_LIMIT = 0.000,
     LO_LO_LIMIT = 0.000,
     PRES_RANGE_MAX = 0.000,
     PRES_RANGE_MIN = 0.000,
     COMPACTION_RATIO = 1.000,
     STORAGE_TYPE = 'STORAGE_TYPE1',
     ALIGNMENT_TIME =  '0s',
     SAMPLE_BLOCKING_RATE =  '5m',
     SAMPLE_INTERVAL =  '6s',
     DB_STORAGE_INT =  '1m',
     EST_LOG_TIME_PER =  '1w'
        end attributes;
```

LOG1

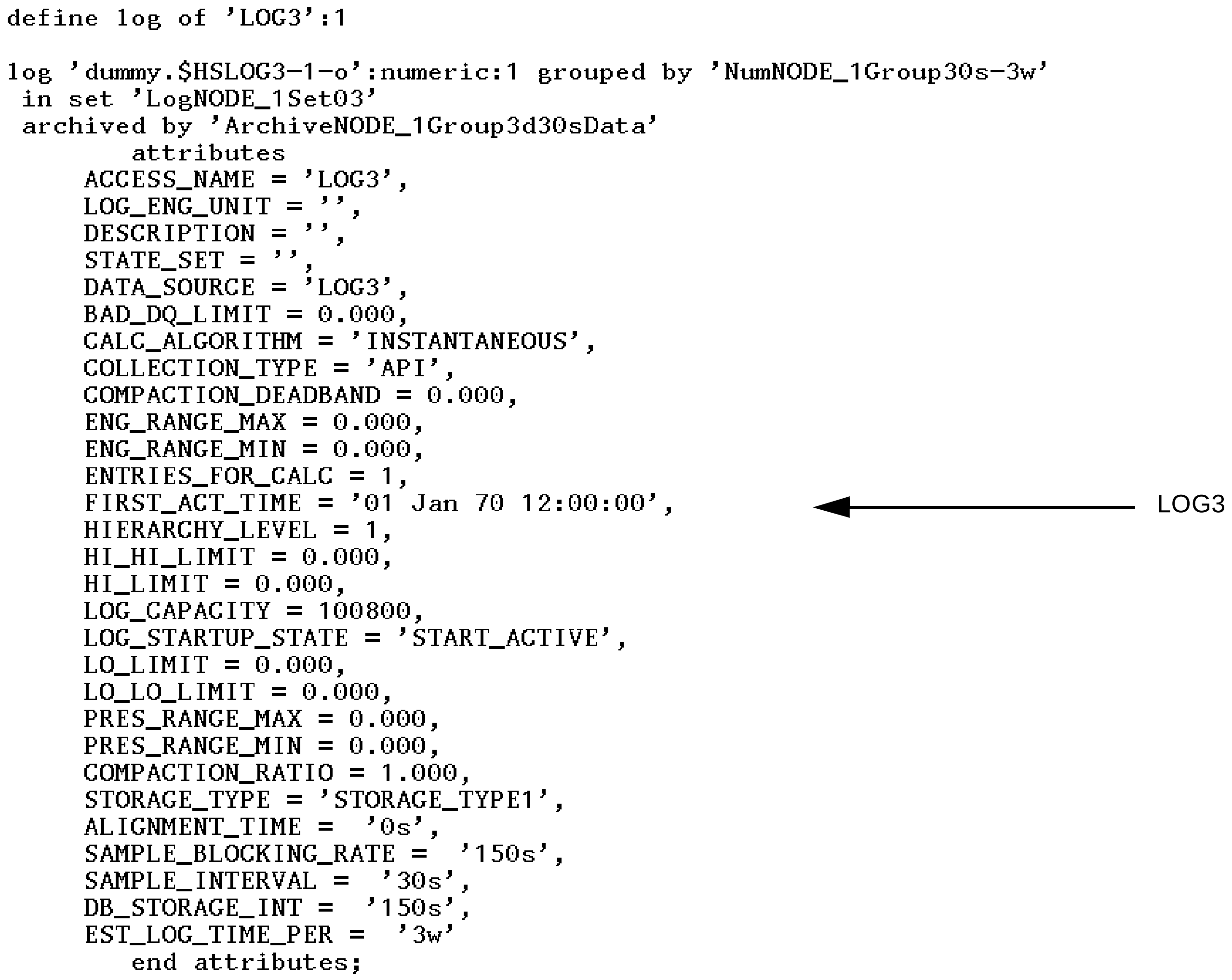
Figure 3-57. Example, dbDump Text File, LOG1



```
define log of ’LOG2’:1

log ’dummy.$HSLOG2-1-o’:numeric:1 grouped by ’NumNODE_1Group15s-3w’
 in set ’LogNODE_1Set02’
 archived by ’ArchiveNODE_1Group3d15sData’
        attributes
     ACCESS_NAME = ’LOG2’,
     LOG_ENG_UNIT = ’’,
     DESCRIPTION = ’’,
     STATE_SET = ’’,
     DATA_SOURCE = ’LOG2’,
     BAD_DQ_LIMIT = 0.000,
     CALC_ALGORITHM = ’INSTANTANEOUS’,
     COLLECTION_TYPE = ’API’,
     COMPACTION_DEADBAND = 0.000,
     ENG_RANGE_MAX = 0.000,
     ENG_RANGE_MIN = 0.000,
     ENTRIES_FOR_CALC = 1,
     FIRST_ACT_TIME = ’01 Jan 70 12:00:00’,
     HIERARCHY_LEVEL = 1,
     HI_HI_LIMIT = 0.000,
     HI_LIMIT = 0.000,
     LOG_CAPACITY = 201600,
     LOG_STARTUP_STATE = ’START_ACTIVE’,
     LO_LIMIT = 0.000,
     LO_LO_LIMIT = 0.000,
     PRES_RANGE_MAX = 0.000,
     PRES_RANGE_MIN = 0.000,
     COMPACTION_RATIO = 1.000,
     STORAGE_TYPE = ’STORAGE_TYPE1’,
     ALIGNMENT_TIME =  ’0s’,
     SAMPLE_BLOCKING_RATE =  ’10m’,
     SAMPLE_INTERVAL =  ’15s’,
     DB_STORAGE_INT =  ’75s’,
     EST_LOG_TIME_PER =  ’5w’
        end attributes;
```

LOG2

Figure 3-58. Example, dbDump Text File, LOG2



```
define log of 'LOG3':1

log 'dummy.$HSLOG3-1-o':numeric:1 grouped by 'NumNODE_1Group30s-3w'
 in set 'LogNODE_1Set03'
 archived by 'ArchiveNODE_1Group3d30sData'
        attributes
     ACCESS_NAME = 'LOG3',
     LOG_ENG_UNIT = '',
     DESCRIPTION = '',
     STATE_SET = '',
     DATA_SOURCE = 'LOG3',
     BAD_DQ_LIMIT = 0.000,
     CALC_ALGORITHM = 'INSTANTANEOUS',
     COLLECTION_TYPE = 'API',
     COMPACTION_DEADBAND = 0.000,
     ENG_RANGE_MAX = 0.000,
     ENG_RANGE_MIN = 0.000,
     ENTRIES_FOR_CALC = 1,
     FIRST_ACT_TIME = '01 Jan 70 12:00:00',
     HIERARCHY_LEVEL = 1,
     HI_HI_LIMIT = 0.000,
     HI_LIMIT = 0.000,
     LOG_CAPACITY = 100800,
     LOG_STARTUP_STATE = 'START_ACTIVE',
     LO_LIMIT = 0.000,
     LO_LO_LIMIT = 0.000,
     PRES_RANGE_MAX = 0.000,
     PRES_RANGE_MIN = 0.000,
     COMPACTION_RATIO = 1.000,
     STORAGE_TYPE = 'STORAGE_TYPE1',
     ALIGNMENT_TIME =  '0s',
     SAMPLE_BLOCKING_RATE =  '150s',
     SAMPLE_INTERVAL =  '30s',
     DB_STORAGE_INT =  '150s',
     EST_LOG_TIME_PER =  '3w'
        end attributes;
```

Figure 3-59. Example, dbDump Text File, LOG3

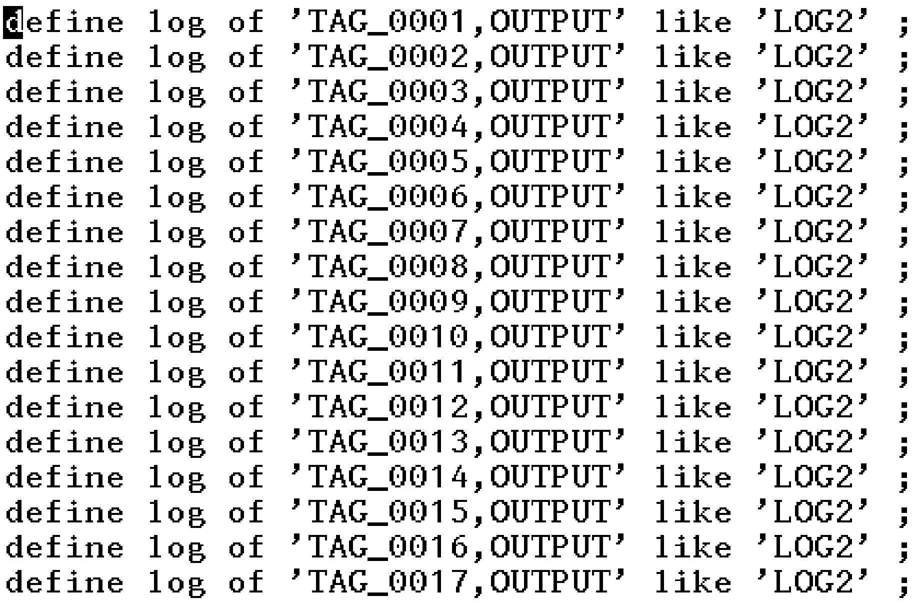
Create Text Files with "Like" Statements to Configure Additional Logs

Use dbDump with the -t option to generate a separate text file with 'like' statements for each category. For example:

```
dbDump -t /tmp/100.tags -a MEASURE -l LOG1 > /tmp/100.like
dbDump -t /tmp/400.tags -a OUTPUT -l LOG2 > /tmp/400.like
dbDump -t /tmp/500.tags -a MEASURE -l LOG3 > /tmp/500.like
```

Enter the name of the text file that contains the tag list to read after the -t option. Enter the attribute to be collected after the -a option. Enter the name of template composite log (defined via the History Builder) that the new logs will be like after the -l option. Enter the name of the text file that will contain the like statements after the > character.

An example of the file produced by this command is shown in Figure 3-60.



```
define log of 'TAG_0001,OUTPUT' like 'LOG2' ;
define log of 'TAG_0002,OUTPUT' like 'LOG2' ;
define log of 'TAG_0003,OUTPUT' like 'LOG2' ;
define log of 'TAG_0004,OUTPUT' like 'LOG2' ;
define log of 'TAG_0005,OUTPUT' like 'LOG2' ;
define log of 'TAG_0006,OUTPUT' like 'LOG2' ;
define log of 'TAG_0007,OUTPUT' like 'LOG2' ;
define log of 'TAG_0008,OUTPUT' like 'LOG2' ;
define log of 'TAG_0009,OUTPUT' like 'LOG2' ;
define log of 'TAG_0010,OUTPUT' like 'LOG2' ;
define log of 'TAG_0011,OUTPUT' like 'LOG2' ;
define log of 'TAG_0012,OUTPUT' like 'LOG2' ;
define log of 'TAG_0013,OUTPUT' like 'LOG2' ;
define log of 'TAG_0014,OUTPUT' like 'LOG2' ;
define log of 'TAG_0015,OUTPUT' like 'LOG2' ;
define log of 'TAG_0016,OUTPUT' like 'LOG2' ;
define log of 'TAG_0017,OUTPUT' like 'LOG2' ;
```

Figure 3-60. Example, Text File with "Like" Statements

Check the Text Files

Use dbLoad to parse and check all text files created via dbDump. For example:

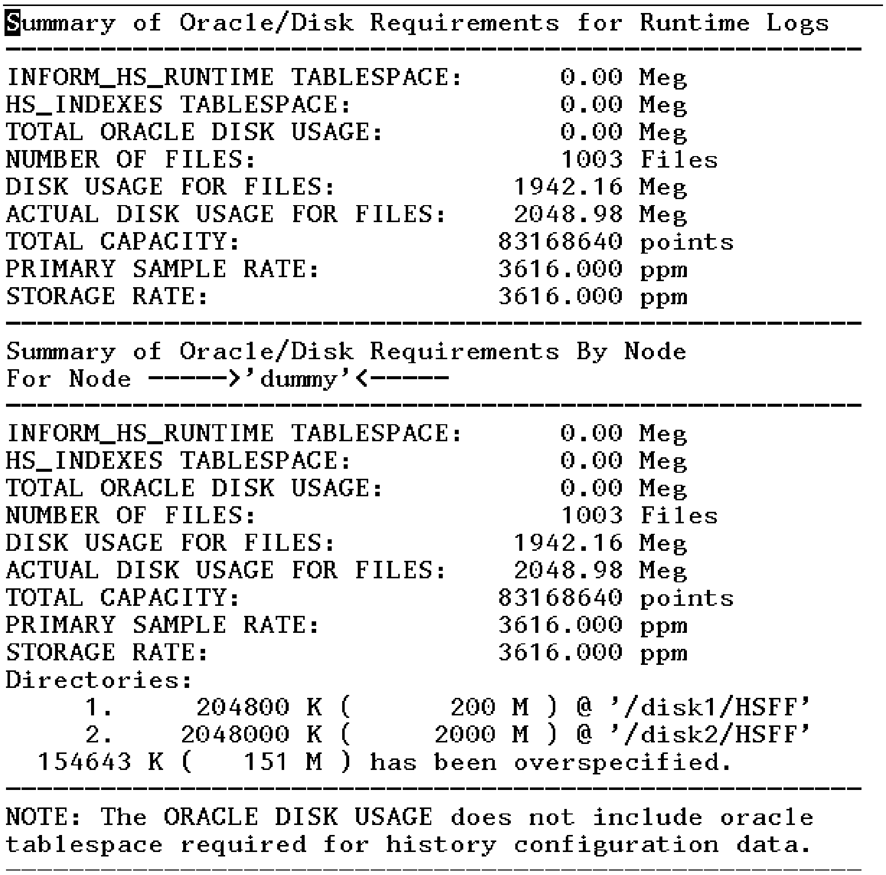
```
dbLoad /tmp/template.db /tmp/100.like /tmp/400.like
/tmp/500.like
```

This generates a report as shown in Figure 3-61. You can direct the report to a text file using the > option. If you do not use this option, the output is directed to the screen.

The report is divided into two parts:

- The top part is labeled Summary of Oracle/Disk Requirements for Runtime Logs. This is an overall summary for all History nodes.
- The bottom part is labeled Summary of Oracle/Disk Requirements by Node. This is a summary of the Oracle/Disk requirements for a specific node. In a dual History configuration there would be two R*equirements by Node* parts. In a single node configuration, this part is essentially a repeat of the first part.

.



```
Summary of Oracle/Disk Requirements for Runtime Logs
------------------------------------------------------
INFORM_HS_RUNTIME TABLESPACE:       0.00 Meg
HS_INDEXES TABLESPACE:              0.00 Meg
TOTAL ORACLE DISK USAGE:            0.00 Meg
NUMBER OF FILES:                    1003 Files
DISK USAGE FOR FILES:            1942.16 Meg
ACTUAL DISK USAGE FOR FILES:     2048.98 Meg
TOTAL CAPACITY:                 83168640 points
PRIMARY SAMPLE RATE:            3616.000 ppm
STORAGE RATE:                   3616.000 ppm
------------------------------------------------------
Summary of Oracle/Disk Requirements By Node
For Node ----->'dummy'<-----
------------------------------------------------------
INFORM_HS_RUNTIME TABLESPACE:       0.00 Meg
HS_INDEXES TABLESPACE:              0.00 Meg
TOTAL ORACLE DISK USAGE:            0.00 Meg
NUMBER OF FILES:                    1003 Files
DISK USAGE FOR FILES:            1942.16 Meg
ACTUAL DISK USAGE FOR FILES:     2048.98 Meg
TOTAL CAPACITY:                 83168640 points
PRIMARY SAMPLE RATE:            3616.000 ppm
STORAGE RATE:                   3616.000 ppm
Directories:
     1.     204800 K (      200 M ) @ '/disk1/HSFF'
     2.    2048000 K (     2000 M ) @ '/disk2/HSFF'
  154643 K (    151 M ) has been overspecified.
------------------------------------------------------
NOTE: The ORACLE DISK USAGE does not include oracle
tablespace required for history configuration data.
------------------------------------------------------
```

Figure 3-61. Example Output from dbLoad

The information in this report is described below:

INFORM_HS_RUNTIME TABLESPACE	Disk space required for History objects stored in the Inform_HS_Runtime tablespace. This includes message logs, asynchronous numeric logs, and SPC logs. Since there are no Oracle-based logs in this example, the Oracle/disk requirement for this component is 0.00 Meg.
HS_INDEXES TABLESPACE	Disk space required for the HS_Indexes tablespace. Since this is related to Oracle-based logs, the Oracle/disk requirement for this component is also 0.00 Meg.
TOTAL ORACLE DISK USAGE	Total disk space required for Oracle tables. This is a sum of the INFORM_HS_RUNTIME and HS_INDEXES fields.
NUMBER OF FILES	Total number of file-based numeric logs.

DISK USAGE FOR FILES	Disk space required for file-based numeric logs before the UNIX overhead is added.
ACTUAL DISK USAGE FOR FILES	Disk space required for file-based numeric logs, including 5% for UNIX overhead.
TOTAL CAPACITY	Sum of log capacity for all logs.
PRIMARY SAMPLE RATE	Average sample rate for all synchronous logs.
STORAGE RATE	Average storage rate for all synchronous logs.
Directories	Allocation of disk space in directories configured for file-based storage. In this example, two directories have been configured: - /disk1/HSFF has been allocated 200 Meg. (204800 K bytes) - /disk2/HSFF has been allocated 2000 Meg. (2048800 K bytes) The line directly below the Directories list indicates whether or not enough disk space has been allocated for the current configuration. In this case, the disk allocation is 151 Meg. more than is required by the History database configuration 2000 Meg. + 200 Meg. - 2048.98 Meg. = 151.02 Meg.

If the dbLoad output indicates that your configuration overloads History resources, you may be able to fix the problem by modifying your History database (change the period of a numeric log and/or decreasing the points per minute sample from the control network). These changes are made to the History database template, and should be relatively easy to make. If your functional requirements are fixed and can not be changed, more hardware will be needed (a second History node, and/or faster machine).

Configure the History Database from the Text Files

If the dbLoad output indicates everything is OK, the database can be loaded:

1. Stop all software on the History node related to the Advant OCS.
2. Drop the current database: `$HS_HOME/bin/hsDBMaint -drop -h`
3. Create an empty database: `$HS_HOME/bin/hsDBMaint -create -h`

   If necessary, use hsDBMaint Extend Tablespace function to extend tablespace, and Directory Maintenance function to increase the quota for file-based logs. The quota should slightly exceed the numbers provided by dbLoad.

4. Use dbLoad with the -l option to load the database. For example:

   ```
   dbLoad -l dummy /tmp/template.db /tmp/100.like /tmp/400.like
   /tmp/500.like
   ```

5. Restart the History node.

   If this load fails because of disk space problems or oracle tablespace problems, just repeat the procedure after you correct the problems.

3.5.13.2 Dual History Example (MOD 300 Only)

Use the History Builder to Configure a Database Template

Refer to the instructions in Section 3.5.13.1, Procedure for Configuring History Database with Text Files for guidelines on how to configure the database template. For dual history, you must configure separate databases on two running history nodes. Configure nodeA, and then configure nodeB with Archive Device(s), Archive Group(s), Log Group(s), Log Set(s) having the same configurations, but different names. The composite logs are built from one node. Use a composite log typical with the structure shown in Figure 3-62.

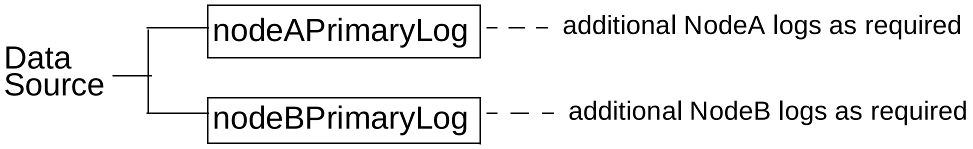

Data Source
nodeAPrimaryLog
additional NodeA logs as required
nodeBPrimaryLog
additional NodeB logs as required

Figure 3-62. Composite Log Typical Structure

To build this composite log typical:

1. Select both nodeA and nodeB from the list of History Managers in the History Configuration window. See Section 3.6.3, History Configuration Window.
2. Create the composite log typical structure with the appropriate log group from each node, and save it to the local node. See Section 3.6.4, Composite Log Typical Window.
3. Create a composite log from the composite log typical created in step 2.

Use a Text Editor to Create Tag Lists for Additional Logs

Also refer to the instructions in Section 3.5.13.1, Procedure for Configuring History Database with Text Files for guidelines on how to create text files for the tag lists.

Read History Database Template into a Text File

Use dbDump with the -d option to dump the template History database to a text file. For example:

on nodeA: `dbDump -d -n nodeA > /tmp/templateA.db`

on nodeB: `dbDump -d -n nodeB > /tmp/templateB.db`

Create Text Files with "Like" Statements to Configure Additional Logs

Use dbDump with the -t option to generate a separate text file with 'like' statements for each category. For example:

```
dbDump -t /tmp/100.tags -a MEASURE -l LOG1 > /tmp/100.like
dbDump -t /tmp/400.tags -a OUTPUT -l LOG2 > /tmp/400.like
dbDump -t /tmp/500.tags -a MEASURE -l LOG3 > /tmp/500.like
```

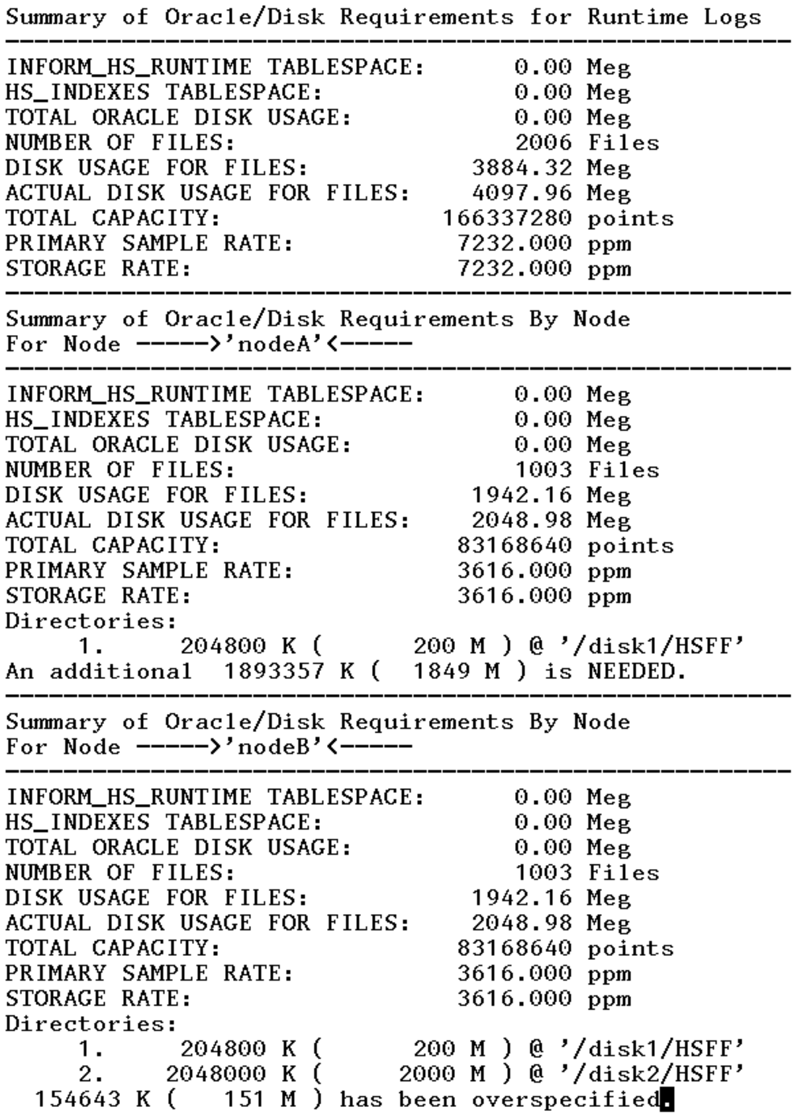
Check the Text Files

Use dbLoad to parse and check all text files created via dbDump. For example:

```
dbLoad /tmp/templateA.db /tmp/templateB.db /tmp/100.like
/tmp/400.like /tmp/500.like
```

This generates a report, Figure 3-63. You can direct the report to a text file using the > option. If you do not use this option, the output is directed to the screen. The report has three parts:

- The top part is an overall summary for all History nodes.
- The middle part is a summary of the Oracle/Disk requirements specifically for nodeA.
- The bottom part is a summary of the Oracle/Disk requirements specifically for nodeB.



```
Summary of Oracle/Disk Requirements for Runtime Logs
------------------------------------------------------
INFORM_HS_RUNTIME TABLESPACE:      0.00 Meg
HS_INDEXES TABLESPACE:             0.00 Meg
TOTAL ORACLE DISK USAGE:           0.00 Meg
NUMBER OF FILES:                   2006 Files
DISK USAGE FOR FILES:           3884.32 Meg
ACTUAL DISK USAGE FOR FILES:    4097.96 Meg
TOTAL CAPACITY:               166337280 points
PRIMARY SAMPLE RATE:           7232.000 ppm
STORAGE RATE:                  7232.000 ppm
------------------------------------------------------
Summary of Oracle/Disk Requirements By Node
For Node ----->'nodeA'<-----
------------------------------------------------------
INFORM_HS_RUNTIME TABLESPACE:      0.00 Meg
HS_INDEXES TABLESPACE:             0.00 Meg
TOTAL ORACLE DISK USAGE:           0.00 Meg
NUMBER OF FILES:                   1003 Files
DISK USAGE FOR FILES:           1942.16 Meg
ACTUAL DISK USAGE FOR FILES:    2048.98 Meg
TOTAL CAPACITY:                83168640 points
PRIMARY SAMPLE RATE:           3616.000 ppm
STORAGE RATE:                  3616.000 ppm
Directories:
     1.     204800 K (      200 M ) @ '/disk1/HSFF'
An additional  1893357 K (  1849 M ) is NEEDED.
------------------------------------------------------
Summary of Oracle/Disk Requirements By Node
For Node ----->'nodeB'<-----
------------------------------------------------------
INFORM_HS_RUNTIME TABLESPACE:      0.00 Meg
HS_INDEXES TABLESPACE:             0.00 Meg
TOTAL ORACLE DISK USAGE:           0.00 Meg
NUMBER OF FILES:                   1003 Files
DISK USAGE FOR FILES:           1942.16 Meg
ACTUAL DISK USAGE FOR FILES:    2048.98 Meg
TOTAL CAPACITY:                83168640 points
PRIMARY SAMPLE RATE:           3616.000 ppm
STORAGE RATE:                  3616.000 ppm
Directories:
     1.     204800 K (      200 M ) @ '/disk1/HSFF'
     2.    2048000 K (     2000 M ) @ '/disk2/HSFF'
  154643 K (    151 M ) has been overspecified.
```

Figure 3-63. Example Report from dbLoad, Dual History

Configure the History Database from the Text Files

If the dbLoad output indicates everything is OK, the database can be loaded:

1. Stop all software on nodeA related to the Advant OCS.
2. Drop the current database: `$HS_HOME/bin/hsDBMaint -drop -h`
3. Create an empty database: `$HS_HOME/bin/hsDBMaint -create -h`

   If necessary, use hsDBMaint Extend Tablespace function to extend tablespace, and Directory Maintenance function to increase the quota for file-based logs. The quota should slightly exceed the numbers provided by dbLoad
4. Use dbLoad with the -l option to load the database on nodeA. For example:

   ```
   dbLoad -l nodeA /tmp/templateA. /tmp/100.like /tmp/400.like
   /tmp/500.like
   ```
5. Restart nodeA.
6. Stop all software on nodeB related to the Advant OCS.
7. Drop the current database: `$HS_HOME/bin/hsDBMaint -drop -h`
8. Create an empty database: `$HS_HOME/bin/hsDBMaint -create -h`

   Again, if necessary use hsDBMaint Extend Tablespace function to extend tablespace, and Directory Maintenance function to increase the quota for file-based logs.
9. Use dbLoad with the -l option to load the database on nodeB. For example:

   ```
   dbLoad -l nodeB /tmp/templateB.db /tmp/100.like
   /tmp/400.like /tmp/500.like
   ```
10. Restart nodeB.

3.5.13.3 dbDump and dbLoad Usage

These utility programs reside in the following directory:

```
/opt/advant/History/bin
```

dbDump

dbDump has two functions. One is to read the history configuration out of the History database and write this configuration to `stdout` (standard output file). To save this information to a file, you must specify the file in the dbDump command. This function uses the -d option described below.

For example:

```
/opt/advant/History/bin/dbDump -d -n testnode1 >
/tmp/dumpfile
```

This command dumps the History database template configured via the History Builder to a text file /tmp/dumpfile on testnode1:

The other function is to create a text file with "like" statements that specify how to configure the bulk of the numeric logs based on template logs configured via the History Builder. This function uses the -t, -a, and -l options.

For example:

```
/opt/advant/History/bin/dbDump -t /tmp/group1.tags -a
MEASURE -l LOG1 > /tmp/group1.like
```

This command creates a text file of 'like' statements for tags listed in the /tmp/group1.tags text file, using the MEASURE (MOD 300) or VALUE (Master) attribute, and the LOG1 composite log template from the History database:

The options for dbDump are described below:

`-d`	Use this option to dump a History database and redirect it to a text file. This option can not be used in combination with -t.
`-t`	Use this option to create a text file with "like' statements that will be used to configure the bulk of the numeric logs in the History database based on the template logs configured via the History Builder.
`-a`	Use this option in conjunction with the -t option to specify the tag attribute for the 'like' statement
`-l`	Use this option in conjunction with the -t option to specify the composite log template for the 'like' statement
`-n` *`nodename`*	Identifies the history node that is being dumped as *`nodename`*. Nodename can be any user defined text and is used by dbLoad to identify the correct objects for a node being loaded. Defaults to *dummy*. This option is only valid in combination with the `-d` option.
> *filename*	Identifies file that history configuration text file is being redirected to.

dbLoad

dbLoad has two functions. One is to check the syntax for the files created by dbDump, and perform dependency checking equivalent to the checks done in the history builder.

For example:

```
/opt/advant/History/bin/dbLoad /tmp/dumpfile
```

NOTE

If database errors are found in the dbLoad file (for instance the value of a high limit attribute is set higher than the high-high limit attribute), the file will not be able to be loaded unless the file is corrected, or the affected logs are deleted from the dbLoad file.

The other function is to load the configuration information from the dbDump text file into a History database identified by the TWO_TASK environment variable.

For example:

```
/opt/advant/History/bin/dbLoad -l node_1 /tmp/dumpfile
```

This command creates objects in the database under node_1.

Usually, dbLoad is used to load data into a clean history database (that is, no other objects have been configured for that node). If objects that already exist in the database are present in the file being loaded, dbLoad will report an error and continue creating other objects in the file.

It is recommended that you remove the old database before loading. Refer to Section 5.6, Database Maintenance Functions. After running dbLoad, you must reboot the History node to establish the new logs in the History Builder.

The syntax for dbLoad is as follows:

```
dbLoad [options] <filename> <filename2>
```

Options:

`-l nodename`	Identifies the history node that is being loaded as *`nodename`*. Nodename must be the nodename provided when the node was dumped using dbDump. The `-l` option tells dbLoad to actually load the file into the database. If `-l` is not provided, no data is loaded; however, the data is checked for validity.
`-v`	Print out the configuration information in the dbDump file. This option will print each object that will be created in the database.
`-a`	Print out the attributes for each object. This option should be used with the `-v` option in order to get detailed attribute information for each object printed with the `-v` option.

3.6 Configuration/Application Menus

This section describes menu items for all AdvaInform History menus related to configuration. These menu items are listed on a menu-by-menu and item-by-item basis. The menus are organized according to the windows where they occur.

Two menus that occur in all AdvaInform History windows are **File** and **Help**. Rather than repeat the descriptions for these menus over and over, they are described just once in Section 3.6.1, Common AdvaInform History Menus.

3.6.1 Common AdvaInform History Menus

3.6.1.1 File Menu

The **File** menu is shown in Figure 3-82.

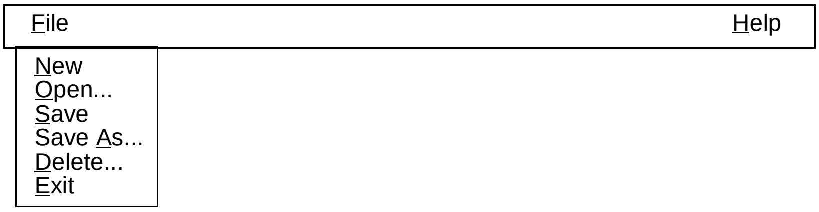


Figure 3-82. File Menu

NOTE

Some menu items shown in Figure 3-82 are not present in the **File** menu on all windows. For instance, the **File** menu for the History Builder and History Configuration windows have just the **Exit** menu item.

File - New

Resets all fields in the current window to the default values so that a new object can be defined.

File - Open

Provides the means to select and display an existing object in its corresponding window. For example, to open a log typical, choose **Open** from the **File** menu in the log typical window. This displays the Open dialog box, Figure 3-83. Select a defined log typical to display it in the window. This functionality is also provided in the Log Set, Log Group, Composite Log Typical, Composite Log, Archive Device, and Archive Group windows.

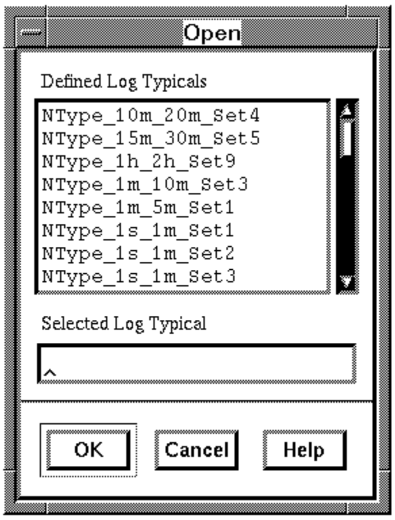


Figure 3-83. Open Dialog for Log Typical Window

File - Save

Stores the currently displayed object into the database and updates the definition of the object.

File - Save As

Makes a copy of the currently displayed object under a new name. Enter the new name in the Save As dialog box, and then click **OK**.

File - Delete

Deletes the currently displayed object.

File - Exit

Exit closes the current window and turns control over to the previous window. You may be prompted to save any unsaved work on the display before the display closes. If you choose **Exit** from the Configuration window, it ends the History Builder session and closes all windows.

3.6.1.2 Help Menu

The **Help** menu is shown in Figure 3-84.

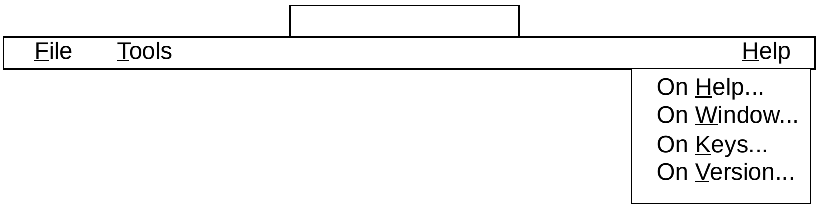


Figure 3-84. History Builder Help Menu Items

Help - On Help

On Help describes how to use the Help feature.

Help - On Window

On Window describes how to use the window that is presently opened. This includes a description of the menu items, fields, and buttons in the display windows and dialog boxes.

Help - On Keys

Help - On Version

On Version provides information about the release of this version of the History Builder including number, date, copyright, and so on.

3.6.2 History Builder Window

The History Builder window is the initial window that is opened when you invoke the **History** menu item in the **AdvaInform** window from the IMS Menu bar, Figure 3-85.

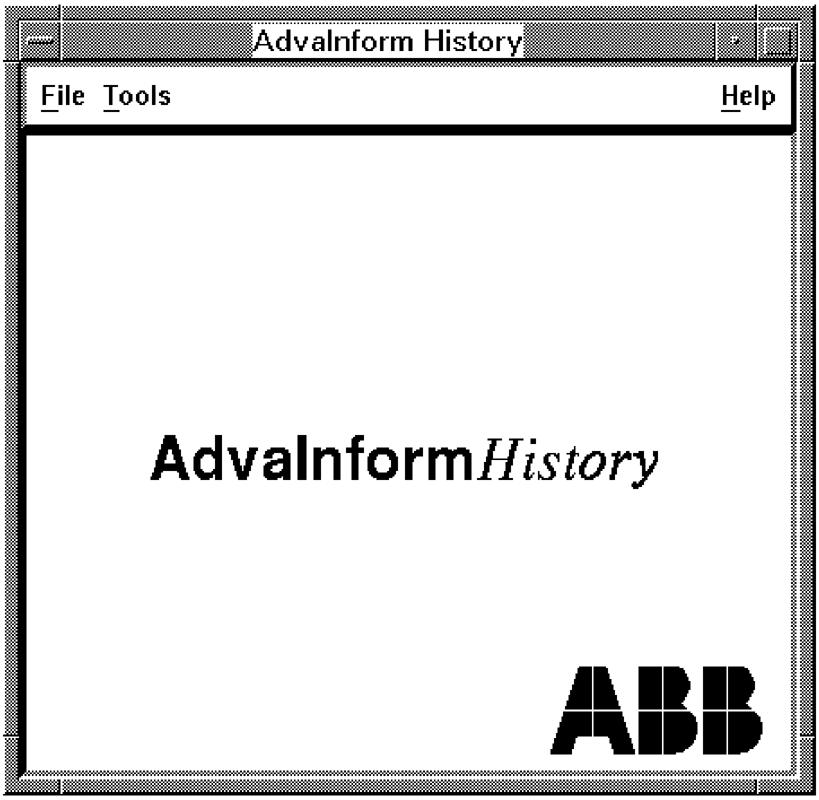


Figure 3-85. AdvaInform History Builder Window

This window provides access to configuration and archive functions, as well as global messages. Global messages are a collection of messages displayed in the message part of all applicable History windows. The menus and corresponding menu items for the History Builder window are shown in Figure 3-86.

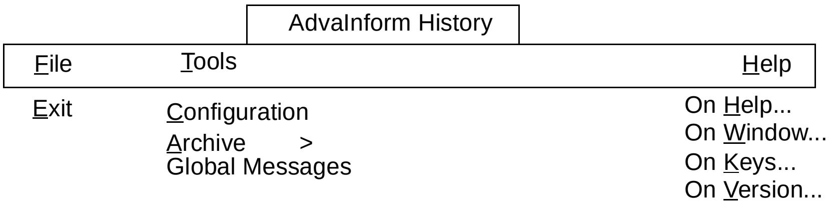


Figure 3-86. History Builder Menu Bar and All Menu Items

3.6.2.1 File Menu

See Section 3.6.1, Common AdvaInform History Menus.

3.6.2.2 Tools Menu

The **Tools** menu in the History Builder is shown in Figure 3-87.

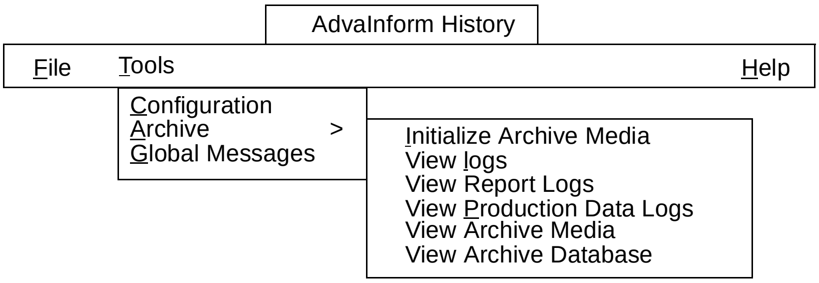


Figure 3-87. History Builder Menu Items

Tools - Configuration

Configuration opens the History Configuration window. Through this window you can select which History Managers you have access to during the current History Builder session. In addition this window provides access to all History configuration tools. The operation of this window is described in Section 3.6.3, History Configuration Window.

Tools - Archive

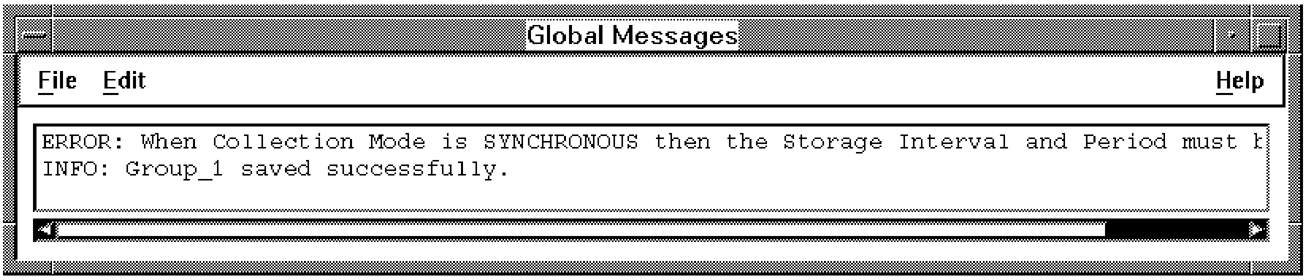
Archive displays a cascade menu as shown in Figure 3-87. Each of these menu items opens a window related to one of the following:

- Initialize Archive Media: Window for initializing a tape loaded in an archive device
- View Logs: A directory of the Numeric and Message Logs contained on long-term storage media
- View Report Logs: A directory of the Report Logs contained on long-term storage media
- View Production Data Logs: A directory of PDLs.
- View Archive Media: A directory of the archived Historical Logs stored on the mounted volume.
- View Archive Database: A directory of the archived Historical Logs in the local Archive database.

These windows are used for archiving historical data during runtime, and are described in detail in Section 4.4.3, Archiving Data for Logs.

Tools - Global Messages

This menu item opens the Global Messages window which lists all ERROR and INFO messages that have been generated for all History windows since the last time messages were cleared, Figure 3-88.



```
ERROR: When Collection Mode is SYNCHRONOUS then the Storage Interval and Period must b
INFO: Group_1 saved successfully.
```

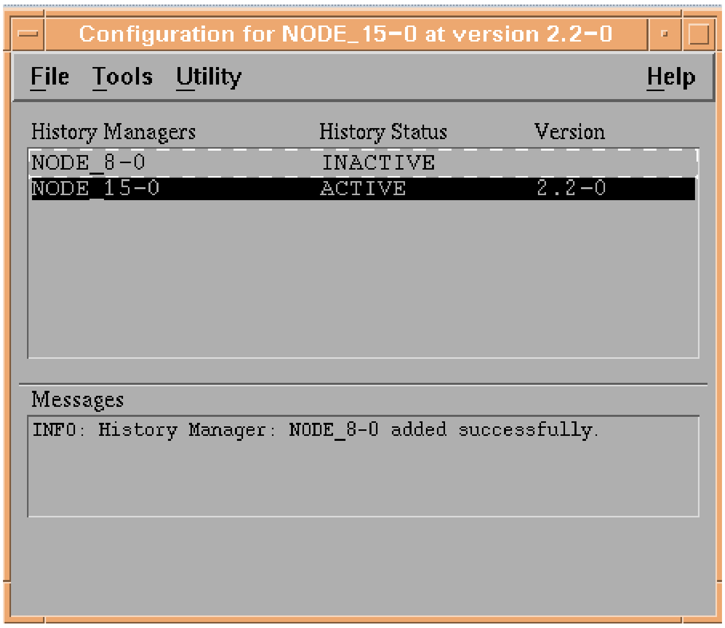
Figure 3-88. Global Messages Window

To clear all messages from this window, choose **Clear** from the **Edit** menu. To close this window, choose **Exit** from the **File** menu.

3.6.3 History Configuration Window

The History Configuration window, Figure 3-89, lets you to select which History Managers to work with during the current History Builder session. In addition, the menus in this window provide access to all History configuration tools.

The purpose and operation of the widgets for this window are described.below.



```
NODE_8-0          INACTIVE
NODE_15-0         ACTIVE        2.2-0
```

```
INFO: History Manager: NODE_8-0 added successfully.
```

Figure 3-89. History Configuration Window

History Managers	The History Managers field lists the local History Manager and any available remote History managers. To work with a History Manager, you must select it as described in Section 3.6.3.1, History Manager Overview
History Status	A History Manager's status is set by the system. The status can be: Active - History Manager is ready to receive data. Inactive - History is on the node, but is not running. Not Available - History is not on the node or the node is unavailable.
Version	This field indicates the version of AdvaInform History that the corresponding History Manager is running.
Messages	The message box displays ERROR and INFO messages that are generated as you use this window. An example of an informative message is: ***INFO: History Manager: NODE_9-0 added successfully

3.6.3.1 History Manager Overview

A History Manager keeps track of the data and the operations performed on that data for all History objects in a node. Such operations include getting the names of objects of a certain type or locating logs related to a particular process object. Each History node has a dedicated History Manager. To configure history objects on a History node, the History Manager for that subsystem must be available to the subsystem you are currently using. It must be running the same version of AdvaInform History software, and it must be active. Loading History onto a node automatically places a History Manager on that node. Only one History Manger can exist on any one node.

When you use the History Builder, the History Manager of the local node is selected by default. For MOD 300, you can select a different History Manager if necessary.If you want to interact with History on any other node, you must acquire access to that node's History Manager.

To interact with a History Manager:

- the History Manager status must be active.
- the database must be at the same version. Version 2.1 cannot access version 1.2 and vice versa.
- the History Manager must be listed in History Configuration window. The local History Manager is always listed. Remote History Managers are added automatically if they are active. For MOD 300, you can add inactive remote History Managers as described in Section 3.6.3.5, Utility Menu (MOD 300).
- you must select the History Manager.

3.6.3.2 Menus for the History Configuration Window

The menus and corresponding menu items for the History Configuration window are shown in Figure 3-90.

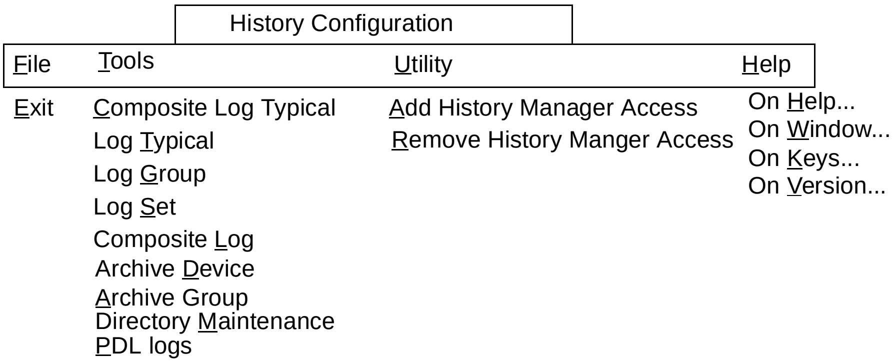


Figure 3-90. History Configuration Menu Bar and All Menu Items

3.6.3.3 File Menu

Exit ends the History Builder session and closes all windows

3.6.3.4 Tools Menu

The **Tools** menu for the Configuration window is shown in Figure 3-91.

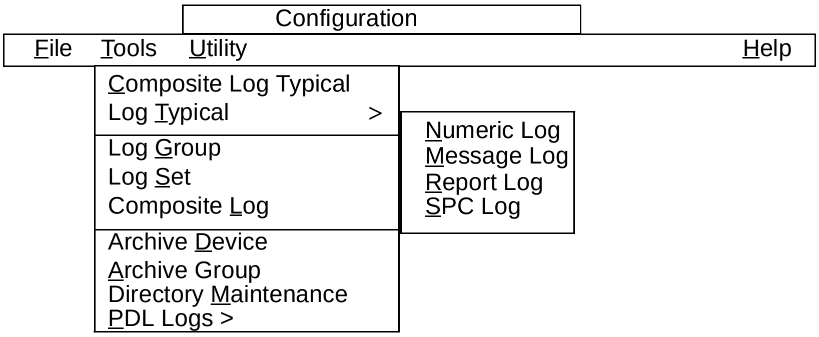


Figure 3-91. Tools Menu Items

Tools - Composite Log Typical

Composite Log Typical opens the Composite Log Typical window for defining Composite Log Typicals. A composite log is a hierarchy of primary and secondary logs. A composite log typical is a re-usable template for creating many similar composite logs. The operation of this window is described in Section 3.6.4, Composite Log Typical Window.

Tools - Log Typical

Log Typical displays a submenu for selecting the log type, Figure 3-92. Numeric, Report, and Message items are always available. SPC is available only if the software package is operating on the History node.

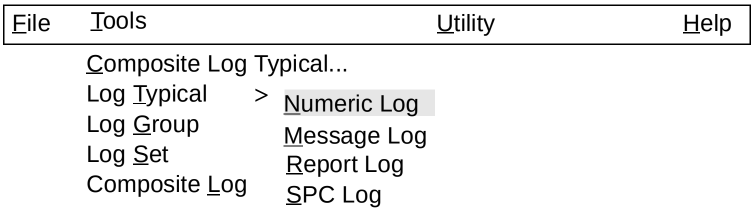


Figure 3-92. Submenu for selecting a Log Typical Type

When you choose one of the Log Typical options, a different version of the Log Typical window is opened. The operation of this window is described in Section 3.6.5, Log Typical Window.

Tools - Log Group

Log Group opens the Log Group window for defining Log Groups. The log group is used to define a group of logs with common storage parameters. Every log must belong to a log group. The operation of the Log Group window is described in Section 3.6.6, Log Group Window.

Tools - Log Set

Log Set opens the Log Set window for defining Log Sets. The log set structure lets you start or stop data collection for a number of logs simultaneously with one command. Log groups also provide this functionality; however, with log sets you have more flexibility regarding logs that you assign to it. The only restriction is that all logs in a log set must reside in the same node. Thus logs in the same log set can have different collection attributes (storage interval, and so on). The operation of this window is described in Section 3.6.7, Log Set Window.

Tools - Composite Log

Composite Log opens the Composite Log window for building a log hierarchy (primary log(s) and associated secondary logs). The operation of this window is described in Section 3.6.8, Composite Log Window

Tools - Archive Device

Archive Device opens the Archive Device window for configuring an Archive Device. The operation of this window is described in Section 3.6.9, Archive Device Window.

Tools - Archive Group

Archive Group opens the Archive Group window. This window is used to group logs for archiving using similar archiving parameters. The operation of this window is described in Section 3.6.10, Archive Group Window.

Tools - Directory Maintenance

Directory Maintenance opens the Directory Maintenance window. This window is used to configure disks and directories for file-based storage of history data. File-based storage applies to logs whose Storage Type attribute is configured to be TYPE1 or TYPE2. The operation of this window is described in Section 3.6.11, Directory Maintenance Window.

Tools - PDL Logs (MOD 300)

This menu item displays a popup menu with two more menu items:

- Delete PDLs - This opens the PDL window which is used to view and delete PDLs and their associated tasks. The operation of this window is described in Section 4.4.4, Handling Production Data Logs (MOD 300).
- Configure PDL Archive - This opens the PDL Archive Configuration window. This window is used to configure how PDLs are archived. The operation of this window is described in Section 3.6.12, PDL Archive Configuration Window (MOD 300).

3.6.3.5 Utility Menu (MOD 300)

The **Utility** menu for the Configuration window is shown in Figure 3-93.

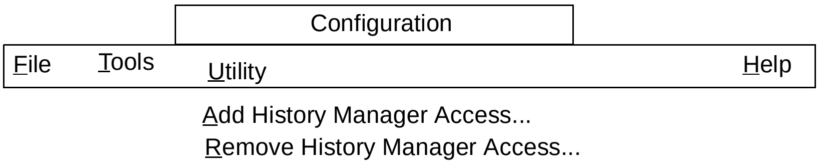
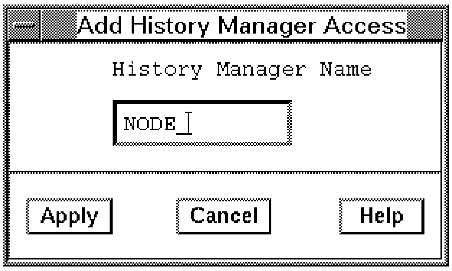

Configuration
File
Tools
Utility
Help
Add History Manager Access...
Remove History Manager Access...

Figure 3-93. Add History Access Menu Item

Utility - Add History Manager Access

Use this menu item to add a remote History Manager that is inactive. The local History Manager, and active remote History managers are automatically added.

When you choose **Add History Manager Access**, the Add History Manager Access dialog box is displayed, Figure 3-94.



History Manager Name

NODE_

Figure 3-94. History Managers Dialog Box

Use this dialog box to enter the name of the History Manager you want to add. The History Manager name for History node in systems with MOD 300 software is:

NODE_n-s

where

n is the DCN node address

s is the sub device (should always be 0)

History Manager name for a History node in systems with Master software is:

NODE_n-s

where

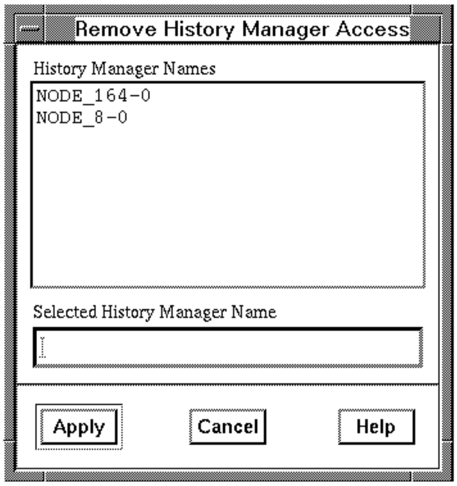
n is the network

s is the node

The prefix part of the History manager name (NODE_) is entered for you. All you need to enter is the DCN node address network and the subdevice node number.

Utility - Remove a History Manager Access

Use this menu item to remove a History Manager from the list on the History Configuration window. When you choose **Remove History Manager Access**, the Remove History Manager dialog box is displayed, Figure 3-95.



NODE_164-0
NODE_8-0

Figure 3-95. Remove History Manager Access Dialog Box

Click on the name of the History Manager to delete and then click on **Apply**. The dialog box remains open so you can delete additional History Managers. You can not delete the local History Manager.

3.6.3.6 Help Menu

See Section 3.6.1, Common AdvaInform History Menus.

3.6.4 Composite Log Typical Window

The Composite Log Typical window is used to define Composite Log Typicals, Figure 3-96. To open the Composite Log Typical window, choose **Composite Log Typical** from the **Tools** menu in the Configuration window.

Composite log typicals are re-usable templates for building two or more similar log hierarchies. To build a composite log typical, you first make a log group for each component log of the hierarchy and then you connect the groups together via the Composite Log Typical window.

For example, the composite log typical in Figure 3-96 has four connected log groups. When you generate a composite log using this composite log typical, the composite log will have a primary log in FC_PRIM_GRP and secondary logs in each of the other groups. These component logs will be connected together in the specified hierarchy.

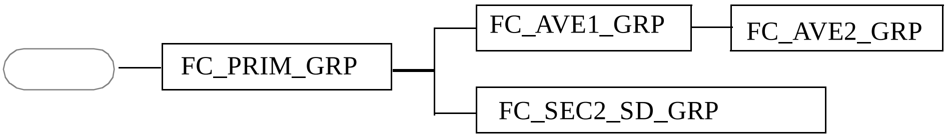


Figure 3-96. Composite Log Typical for Building Flow Log Hierarchy

NOTE

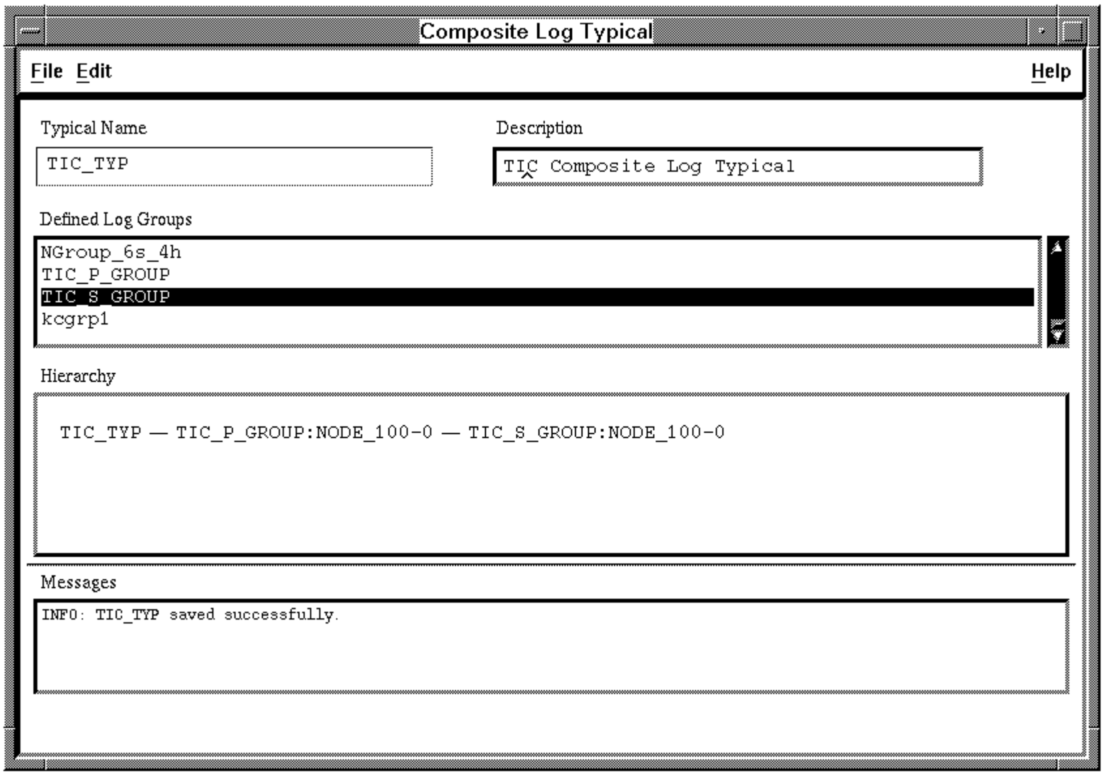
Before you build a Composite Log Typical, the log groups you use to build it must already exist. Refer to Section 3.6.6, Log Group Window.

When you define a log hierarchy, there are relationships which must be maintained between a log which receives input from another log and the source log which provides the input. The receiving log is always a secondary log. The source log may be a primary log, or another secondary log. The relationships are as follows:

- The sample interval for the receiving log must be a multiple of the storage interval for the source log.
- The sample interval for the receiving log must be greater than or equal to the storage interval for the source log.
- The time period for the receiving log must be greater than or equal to the time period for the source log.
- A composite log can have both synchronous and asynchronous primary logs. However, an asynchronous log can only be a primary log, and can not be the data source for a secondary log.

When you open this window initially, the fields are undefined and there is a list of configured log groups.

The purpose and operation of the widgets in this window are described below.



Typical Name
TIC_TYP

Description
TIC Composite Log Typical

Defined Log Groups
NGroup_6s_4h
TIC_P_GROUP
TIC_S_GROUP
kcgrp1

Hierarchy
TIC_TYP — TIC_P_GROUP:NODE_100-0 — TIC_S_GROUP:NODE_100-0

Messages
INFO: TIC_TYP saved successfully.

Figure 3-97. Composite Log Typical Window

Typical Name	When you configure a new composite log typical, use this field to enter the name.
Description	Use this field to enter a description for the Composite Log Typical. This description is optional and for information only.
Log Groups	Shows a current list of log groups on the node. You can select log groups to include in the composite log typical via this field.
Hierarchy	Shows the hierarchy of the Composite Log Typical.
Messages	Displays any error/information messages.

The procedure for building a composite log typical is demonstrated in Section 3.4.3.3, Build a Composite Log Typical.

The menus for the Composite Log Typical Window are shown in Figure 3-98.

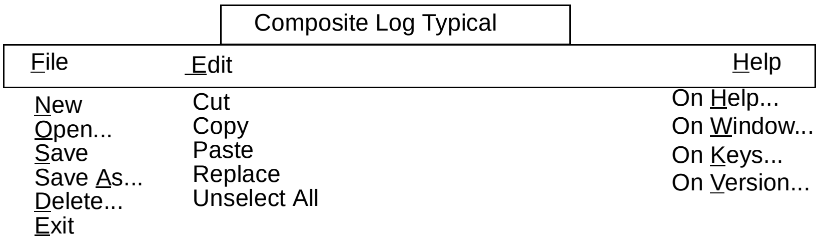


Figure 3-98. Composite Log Typical Menu Bar Items

3.6.4.1 File Menu

See Section 3.6.1, Common AdvaInform History Menus.

3.6.4.2 Edit Menu

Use the **Edit** menu, Figure 3-99, to make changes to the information that has been entered into the text field in the Composite Log Typical display window.

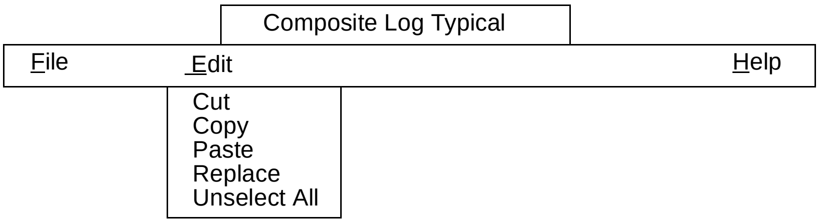


Figure 3-99. Composite Log Typical Edit Menu Items

Edit - Cut

Deletes the selected Log Group(s) from the hierarchy and stores them in the paste buffer. This cuts all secondary Logs connected to the selected Log Group(s).

Edit - Copy

Puts a duplicate copy of the currently selected Log Group(s) in the paste buffer.

Edit - Paste

Inserts the last copied or cut Log Group(s) after the currently selected Log Group(s).

Edit - Replace

Replaces the selected Log Group(s) in the hierarchy with the selected Log Group from the Log Groups list.

Edit - Unselect All

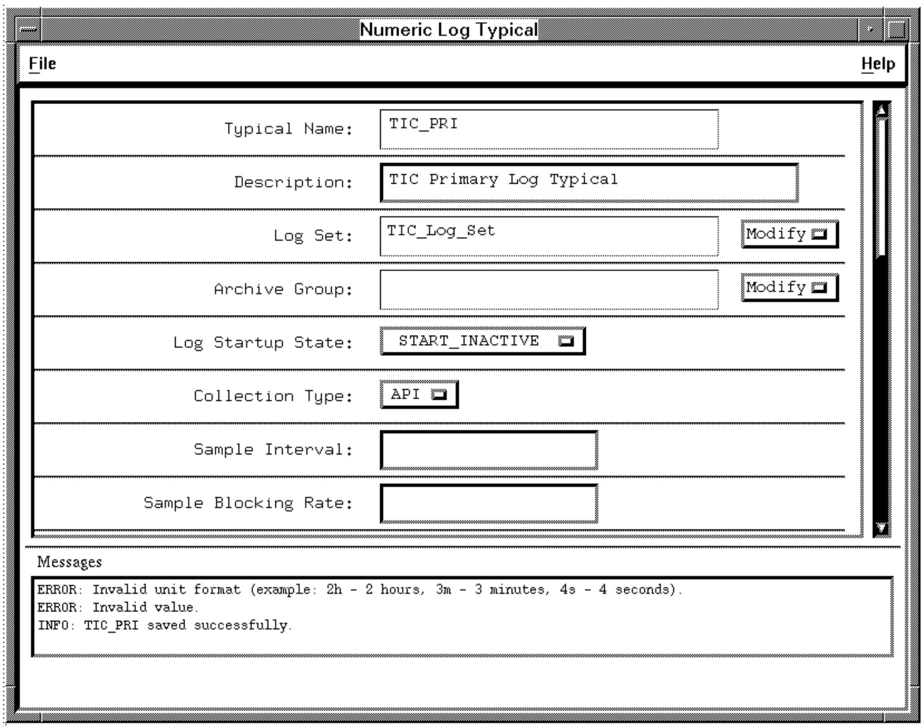
Cancels the selection of all Log Groups.

3.6.4.3 Help Menu

See Section 3.6.1, Common AdvaInform History Menus.

3.6.5 Log Typical Window

When you choose **Log Typical** from the **Tools** menu in the History Configuration window, and then choose a log type from the pop-up menu, the appropriate Log Typical edit window is opened. For example, Figure 3-100 shows the edit window for Numeric Log Typicals. Windows for other log typicals have the same format with different attributes.



Typical Name: TIC_PRI
Description: TIC Primary Log Typical
Log Set: TIC_Log_Set
Archive Group:
Log Startup State: START_INACTIVE
Collection Type: API
Sample Interval:
Sample Blocking Rate:

```
ERROR: Invalid unit format (example: 2h - 2 hours, 3m - 3 minutes, 4s - 4 seconds).
ERROR: Invalid value.
INFO: TIC_PRI saved successfully.
```

Figure 3-100. Example Log Typical Window (Numeric)

When you open this window initially, the fields are undefined or at their default values. The attributes for each particular log type are described in Section 3.1, Design Considerations.

In addition to building new log typicals, you can open existing log typicals and delete existing log typicals. This functionality is provided via menu items in the **File** menu in the Log Typical Window.

The procedure for building a log typical is demonstrated in Section 3.4.3.1, Build Log Typicals.

3.6.6 Log Group Window

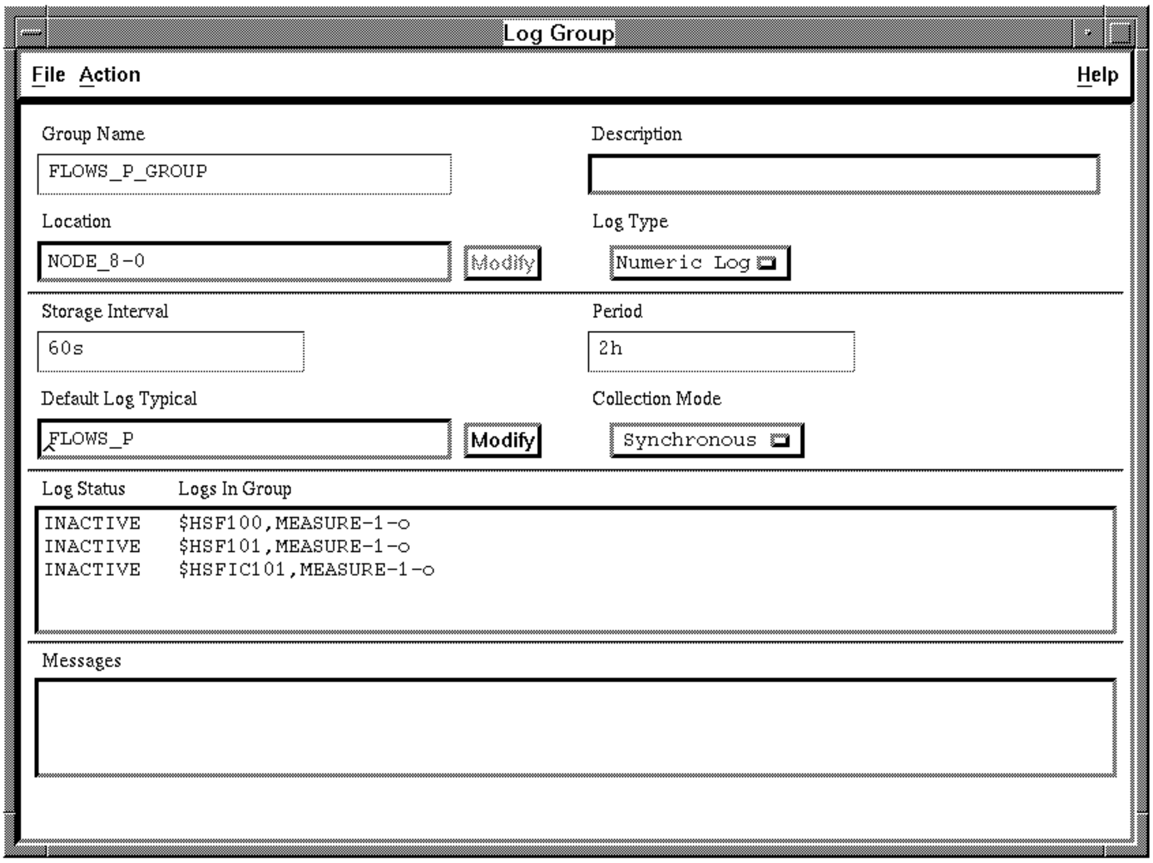
The Log Group is a structure for grouping logs with common storage parameters. To open the Log Group window, choose **Log Group** from the **Tools** menu in the Configuration window.

Every log must belong to a log group. A log group specifies:

- the history node where the log resides.
- the log type such as numeric or message.
- basic data collection attributes including storage interval, collection mode, and log period.
- As an option, a log group can reference a log typical which is a template for many similar logs (attribute values are configured the same).

All logs in a log group reside on the same node, they are all of the same log type, and they all have the same collection mode, storage interval, and period. Although the log period is initially defined on a group basis, it can be changed on an individual basis for logs within a group. You must build log groups before you build logs. Build as many log groups as required.

The purpose and operation of the widgets in this window are described below.



Group Name: FLOWS_P_GROUP

Description:

Location: NODE_8-0

Log Type: Numeric Log

Storage Interval: 60s

Period: 2h

Default Log Typical: FLOWS_P

Collection Mode: Synchronous

Log Status	Logs In Group
INACTIVE	$HSF100,MEASURE-1-o
INACTIVE	$HSF101,MEASURE-1-o
INACTIVE	$HSFIC101,MEASURE-1-o

Messages

Figure 3-101. Log Group Display Window

Group Name	When you configure a new log group, use this field to enter the log group name.
Description	Use this field to enter a description for the log group (optional).
Location	Use this field to specify the location (node) where the log group will reside.Click on the **Modify** button by this field to display a list of nodes to choose from. **Modify** is deactivated when you open an existing Log Group.
Log Type	Use this field to specify the log type for the log group. Press the target on the right side of this field to display a list of log types to choose from. All logs in a log group are the same log type. This button is deactivated when you open an existing Log Group.
Storage Interval	Use this field to specify the storage interval for the log group. Enter an integer value ≤ 4095 with time unit indicator: w = week, d = day, h = hour, m = minute, s = second. All logs in the log group have the same storage interval. This field is locked if you open an existing log group. To store all values at the end of the month, enter: **`monthly`** for storage interval.
Period	Use this field to specify the minimum time span for the log group Enter an integer value ≤ 4095 with time unit indicator: w = week, d = day, h = hour, m = minute, s = second. All logs in the log group are initially configured the same period. This field is locked if you open an existing log group.
Default Log Typical	This field displays the name of the default log typical for the log group. You can change the default log typical.Click on the **Modify** button to display a list of log typicals to choose from. You can leave this field blank in which case the logs have default values for their attributes.
Collection Mode	Use this field to specify the collection mode for this log group. The mode can be synchronous or asynchronous. If synchronous, you have to specify the storage interval and period. You do not have to specify these attributes if the collection mode is asynchronous. All logs in the log group have the same collection mode
Logs in Group	This field displays the names of the logs that currently exist in the group and their state. This is an information only field. You do not make entries into this field. Logs are added to the log group via the Composite Log window.
Messages	Displays error and information messages.

The procedure for building logs groups is demonstrated in Section 3.4.2.1, Configure a Log Group.

The menus for the Log Group window are shown in Figure 3-102.

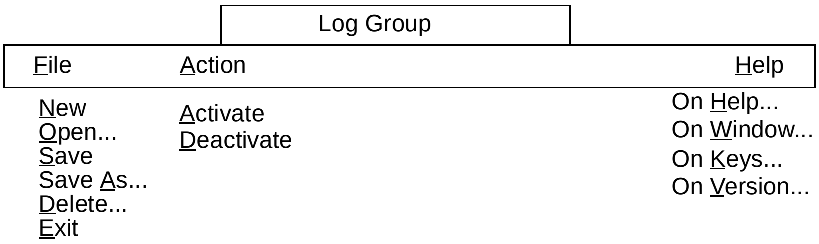


Figure 3-102. Log Group Menu Bar

3.6.6.1 File Menu

See Section 3.6.1, Common AdvaInform History Menus.

3.6.6.2 Action Menu

Use the **Action** menu, Figure 3-103, to select the status of a particular Log Group.

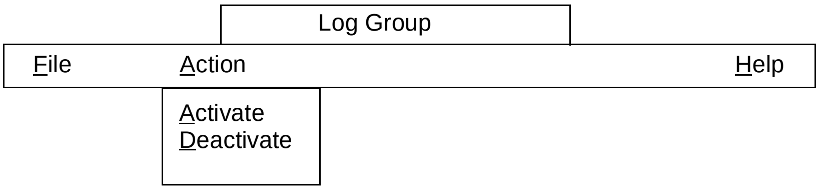


Figure 3-103. Action Menu Item

Activate

Start data collection for the logs in the log group.

Deactivate

Stop data collection for the logs in the log group.

3.6.6.3 Help Menu

See Section 3.6.1, Common AdvaInform History Menus.

3.6.7 Log Set Window

The Log Set window is used to define Log Sets, Figure 3-104. The log set structure lets you start or stop data collection for a number of logs simultaneously with one command. Log groups also provide this functionality; however, with log sets you have more flexibility in the logs that you assign to it. The only restriction is that all logs in a log set must reside in the same node. Thus logs in the same log set can have different collection attributes (storage interval, and so on).

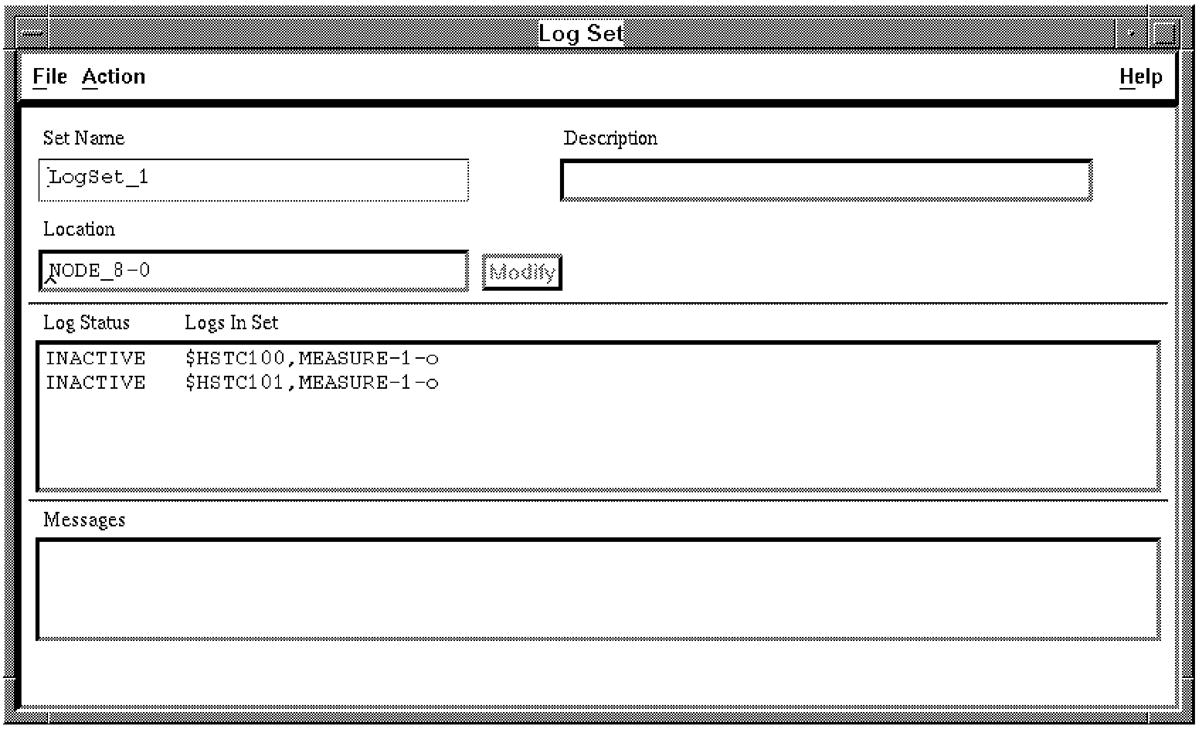
For MOD 300, you can configure up to 16 log sets that can be Activated/Deactivated via the RECORDER statement in a TCL program. These log sets must be named according to the following format:

`TCLLOGSET`*n*

where *n* = an integer from 1 to 16. You can configure more than 16 log sets; however, only file sets named according to the above format can be controlled via the RECORDER statement.

It is recommended that you build log sets prior to building the logs. However, if you build logs first you can still add log sets later and assign the existing logs to them. In addition to building new log sets, you can open an existing log set, and delete existing log sets. This functionality is provided via menu items in the **File** menu in the Log Set Window.

To open the Log Set Window, choose **Log Set** from the **Tools** menu in the Configuration window. When you open this window initially, the fields are undefined. The purpose and operation of the widgets in this window are described below.



Set Name: LogSet_1

Description:

Location: NODE_8-0

```
INACTIVE    $HSTC100,MEASURE-1-o
INACTIVE    $HSTC101,MEASURE-1-o
```

Figure 3-104. Log Set Display Window

Set Name	When you configure a new Log Set, use this field to enter the set name. Log sets to be controlled by the RECORDER statement in a TCL program must be named using the format **`TCLLOGSET`***n* where n is an integer from 1 to 16.
Description	Use this field to enter a description for a log set (optional).
Location	Use this field to specify the location (node) for a log set. Click on the **Modify** button to display a list of nodes to choose from. This button is deactivated when you open an existing Log Set
Logs in Set	This field displays a list of Logs in the currently displayed set and their state. This is a read only field. You cannot make entries to this field.

The menu bar for the Log Set window is shown in Figure 3-105.

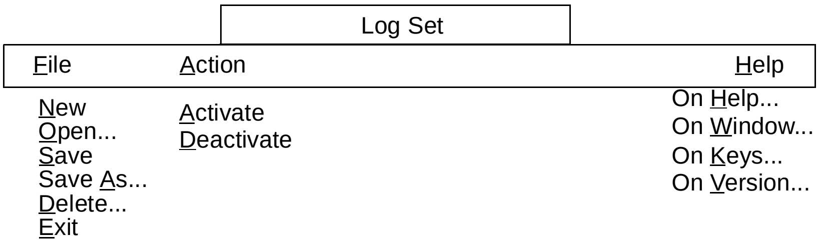


Figure 3-105. Log Set Menu Bar

3.6.7.1 File Menu

See Section 3.6.1, Common AdvaInform History Menus.

3.6.7.2 Action Menu

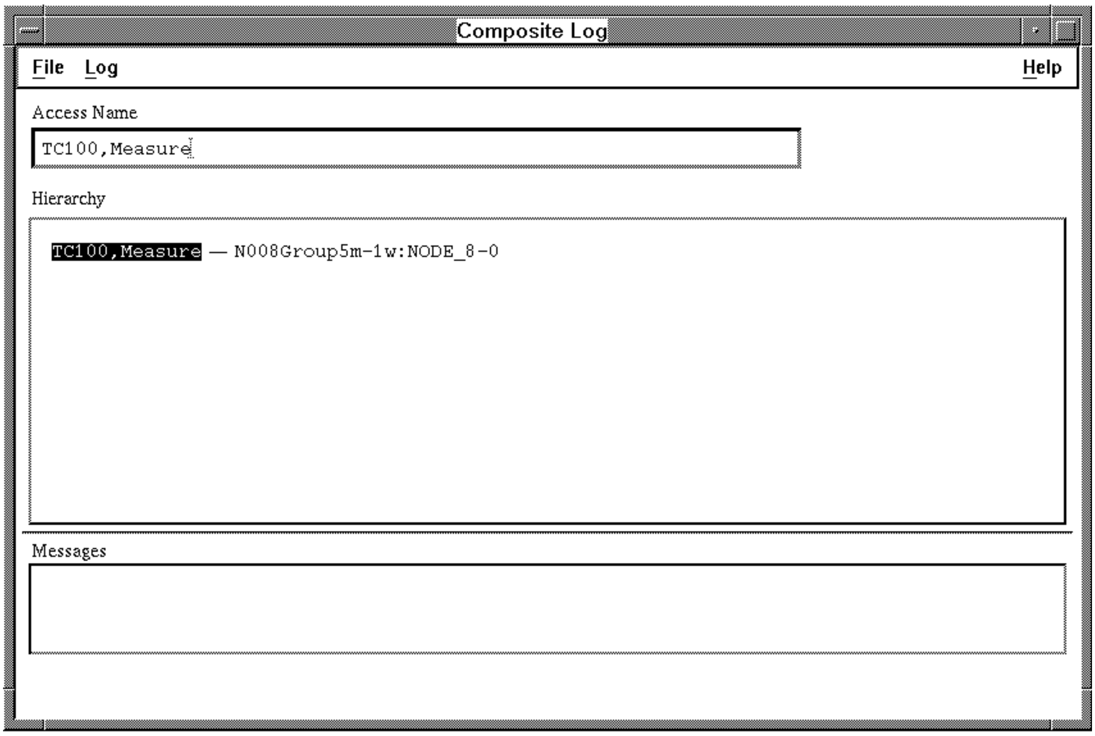
See Section 3.6.6.2, Action Menu.

3.6.7.3 Help Menu

See Section 3.6.1, Common AdvaInform History Menus.

3.6.8 Composite Log Window

The Composite Log window is used to build a log hierarchy, Figure 3-105. To open the Composite Log window, choose **Composite Log** from the **Tools** menu in the Configuration window. When you open this window initially, the fields are undefined. The widgets in this window are described below.



Access Name
TC100,Measure

Hierarchy
TC100,Measure — N008Group5m-1w:NODE_8-0

Messages

Figure 3-106. Composite Log Typical Window

Access Name	Use this field to specify the data source for a log (by object name and attribute name). The entry format is *object, attribute*. Use all uppercase letters when you enter the attribute name. For example: TC100, MEASURE or TC100,VALUE For an FCM attribute a period separates the tag and FCM name. for example: FC100.PID, RESULT The access name must match the data source name and attribute. If they do not match, you will not be able to retrieve log data using the name of the process object. Also, History will not be able to cascade CCF trend, engineering units, alarm, and presentation attributes from the data source to the log.
Log Hierarchy	Use this field to specify the log hierarchy. First select the log name to highlight it, and then click the right mouse button. This displays a pop-up menu. To build a simple log, choose, **Add Log** from the menu to display a list of allowable log groups to add. Then select a log group from the list. To build a composite log, choose **Add Comp Log Typ** from the menu to display a list of available composite log typicals. Then select a composite log typical from the list. You can also build a composite log without a composite log typical by clicking on the primary log group with the right mouse button. This displays a pop-up menu with the **Add Log** menu item as described above.
Messages	Displays error and information messages.

Two detailed examples of configuring composite logs are provided in Section 3.4, Tutorial. Section 3.4.2.2, Build a Basic Composite Log shows how to build a composite log without using typicals. Section 3.4.3.4, Build Composite log using a Composite Log Typical show how to build a composite log using a composite log typical.

The menus for the Composite Log window are shown in Figure 3-107.

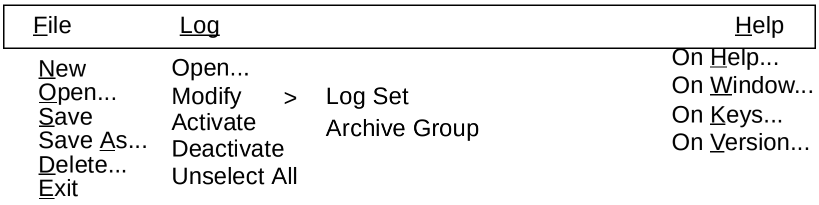


Figure 3-107. Composite Log Menu Bar

3.6.8.1 File Menu

See Section 3.6.1, Common AdvaInform History Menus.

3.6.8.2 Log Menu

Log is selected to gain access to a particular log in the Composite Log. These items are only available when a selection(s) are made in the hierarchy.

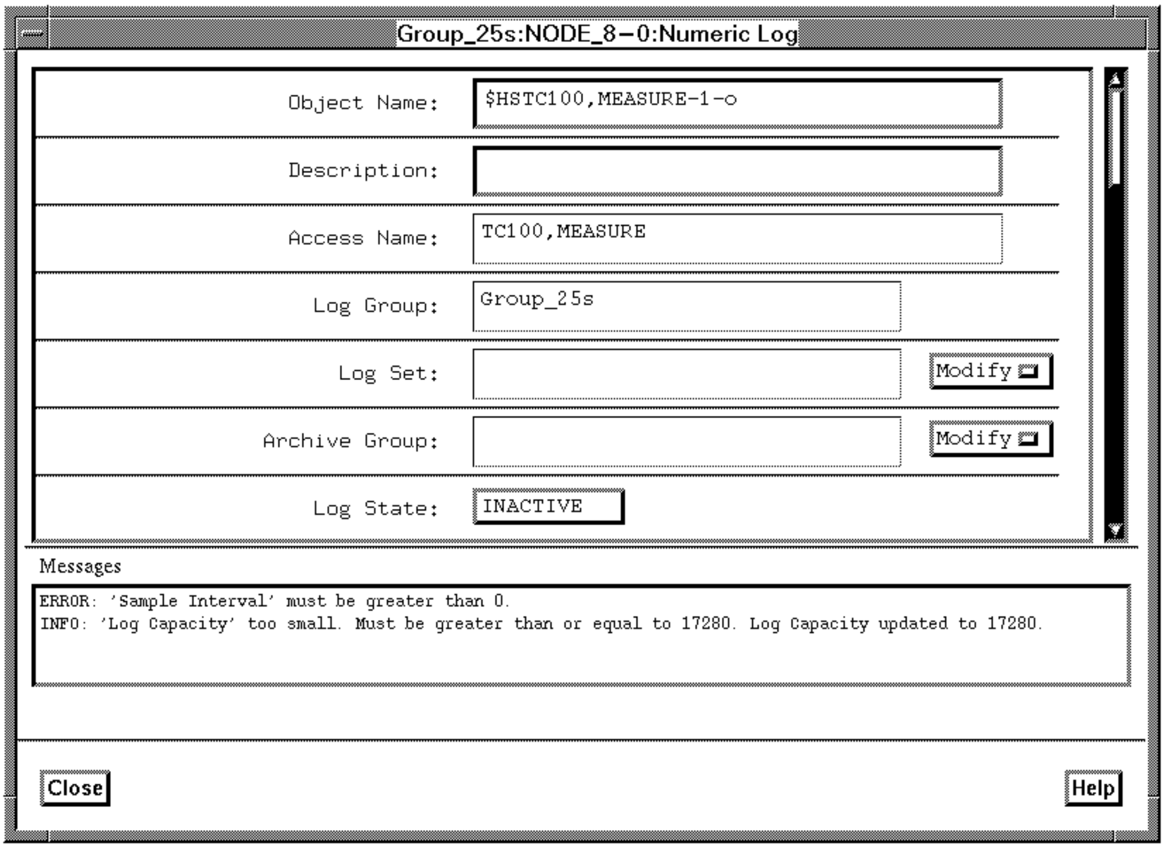
Log - Open

Open opens the Log Attribute edit window for the Logs selected in the hierarchy, Figure 3-108. This window lets you change the log's attributes. First select a Log in the hierarchy, then choose **Open** from the **File** menu.

NOTES

> You can not click on the data source name in the hierarchy, but rather, the log group name.
>
> To open more than one log at a time, hold down the left mouse button and drag to select the logs that you want to open.

The Log Attribute dialog box is similar to the Log Typical display window. This dialog box displays all the attributes of the log.


Group_25s:NODE_8–0:Numeric Log

Object Name: $HSTC100,MEASURE-1-o
Description:
Access Name: TC100,MEASURE
Log Group: Group_25s
Log Set:
Archive Group:
Log State: INACTIVE

Messages

```
ERROR: 'Sample Interval' must be greater than 0.
INFO: 'Log Capacity' too small. Must be greater than or equal to 17280. Log Capacity updated to 17280.
```

Figure 3-108. Log Attribute Edit Window

Log - Modify

When you choose **Modify,** a secondary menu is displayed with the following items: **Log Set**, and **Archive Group**. Each item is described below.

Log Set	Use this menu item to assign the selected log to a log set. When you choose this menu item, the Modify Log Set dialog box is displayed, Figure 3-49. Select the log set and then click on **OK**.
Archive Group	Use this menu item to assign the selected log to an Archive Group. When you choose this menu item, the Modify Archive dialog box is displayed, Figure 3-38. Select the Archive Group and then click on **OK**.

Log - Activate

This activates logs selected in the hierarchy. To activate more than one log at a time, hold down the left mouse button and drag to select the logs that you want to activate.

Log - Deactivate

This deactivates logs selected in the hierarchy. To deactivate more than one log at a time, hold down the left mouse button and drag to select the logs that you want to deactivate.

Log - Unselect All

This option is available if a log is selected. **Unselect All** cancels the selection of all selected Log Groups.

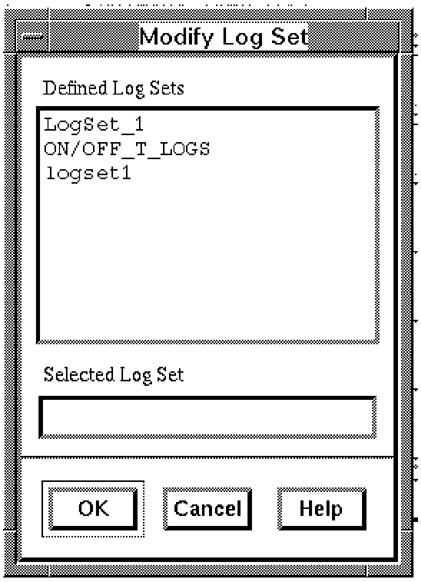


Figure 3-109. Modify log Set Dialog Box

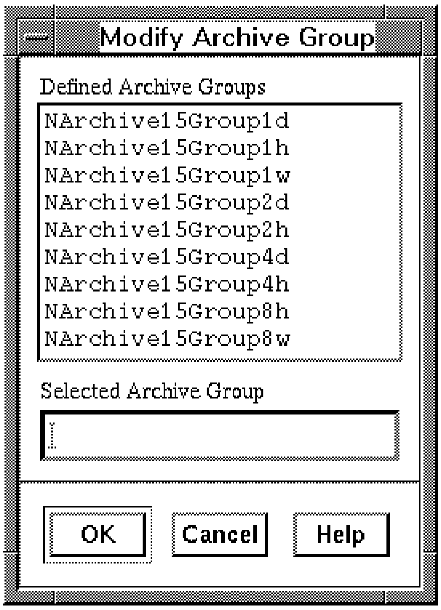


Figure 3-110. Modify Archive Group Dialog Box

Log - ReCalculateLimits

This menu item is only applicable if you use the AdvaInform SPC option in conjunction with AdvaInform History.

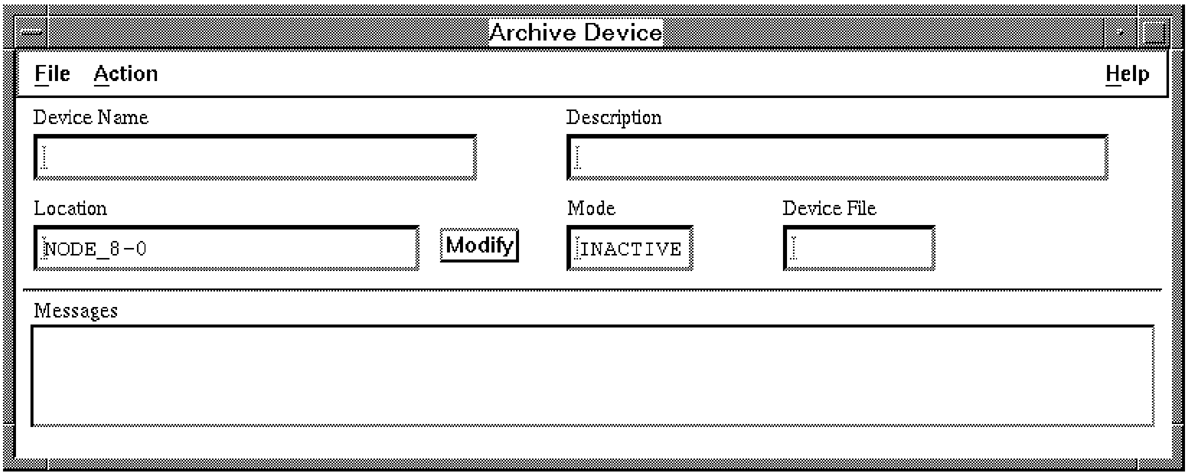
SPC limits can either be calculated or specified by the user. When calculated, History calculates the limits based on the data collected for the SPC log. When enough data has been collected, the limits are calculated and stored in the data base. The limits are not updated unless you specify that they be recalculated via this menu item.

To recalculate limits, first select the applicable SPC log in the Composite Log window, and then choose **RecalculateLimits** from the **Log** menu. As an alternative, you can select the SPC log in the Composite Log window, and then click the right mouse button.

Only one SPC log can be selected at a time.

3.6.9 Archive Device Window

The Archive Device window is used to define archive devices, Figure 3-111. To open this window, choose **Archive Devic**e from the **Tools** menu in the Configuration window.



Location: NODE_8-0

Mode: INACTIVE

Figure 3-111. Archive Device window

The archive device is the DAT tape drive where you store archived logs. You can restore the archived logs from the archive device to the History node internal drive as well. Before you configure the archive device, the device must be connected to the History node as described in the *Advant Station Hardware User's Guide* and set up as described in the *Advant Station 500 Series IMS User's Guide*.

When you open this window initially, the fields are undefined. When you open this window for an existing Archive Device, the corresponding information is displayed. The purpose and operation of the widgets for this window are described below.

Device Name	This is the name you use to identify the device when you specify where to archive data to, or where to retrieve archive information from.
Description	This is an optional description to further identify the archive device.
Location	This is the name of the History manager for the node that the archive device is connected to. Press the **Modify** button next to this field to display a list of History Managers.
Mode	The Mode can be ACTIVE or INACTIVE. The device must be active in order to archive data to, or restore data from the device. The default mode is ACTIVE. You can not change the mode directly in this field. To change the mode, use the **Action** menu.

Device File	The archive device has pre-defined device files. Specify device files for archival as they were named when the device file was created. Typically, there is a device file named **`0m`**. This file is set up such that the archive device will rewind automatically after each archive function. **3m** is another common device file name. DO NOT enter mn. This causes the tape to not rewind after it has been accessed, and prevents archive from functioning properly.
Messages	This field displays messages related to activities you do in the Archive Device window such as error messages.

The menus for the Archive Device window are shown in Figure 3-112.

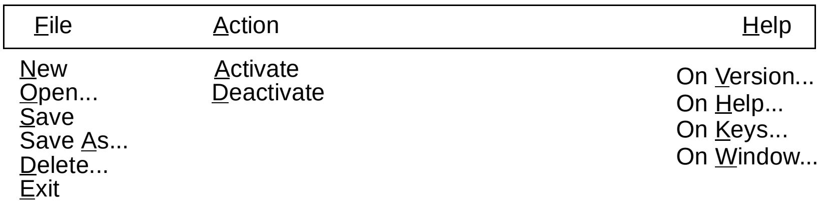


Figure 3-112. Archive Device and archive Group Menu Bar

3.6.9.1 File Menu

See Section 3.6.1, Common AdvaInform History Menus.

3.6.9.2 Action Menu

Action - Activate

Activate the archive device (or group).

Action - Deactivate

Deactivate the archive device (or group).

3.6.9.3 Help Menu

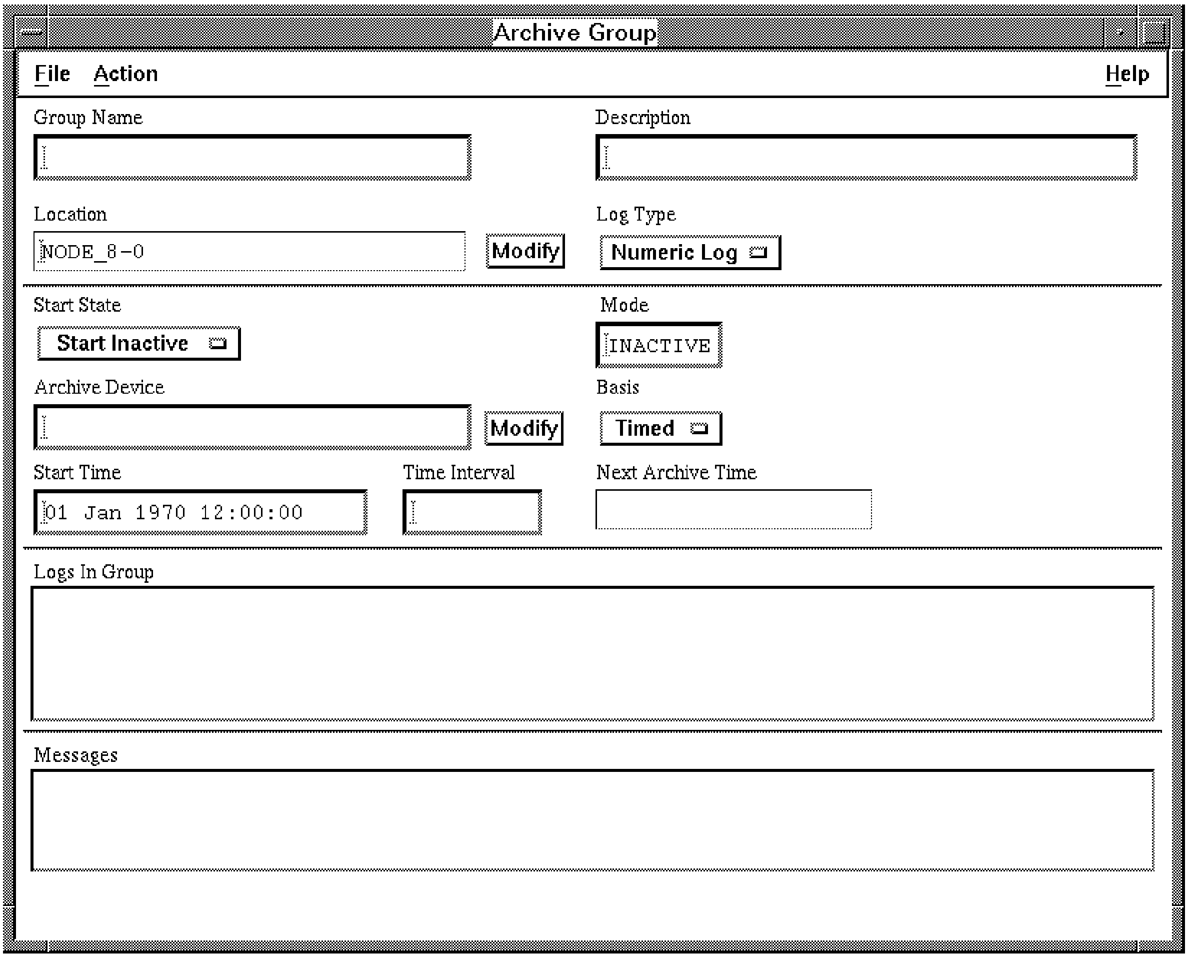
See Section 3.6.1, Common AdvaInform History Menus.

3.6.10 Archive Group Window

The Archive Group window is used to configure archive groups, Figure 3-113. To open this window, choose **Archive Group** from the **Tools** menu in the Configuration window.

An archive group is a group of logs that you configure such that you can treat the logs as a unit for archival, rather than deal with the logs independently. You configure archive groups via the Archive Group window.

When you open this window initially, the fields are undefined. When you open this window for an existing Archive Group, the corresponding information is displayed. The purpose and operation of the widgets for this window are described below.



Location: NODE_8-0
Log Type: Numeric Log
Start State: Start Inactive
Mode: INACTIVE
Archive Device
Basis: Timed
Start Time: 01 Jan 1970 12:00:00
Time Interval
Next Archive Time

Figure 3-113. Archive Group Window

Group Name	This is the name used to identify the archive group when you specify an archive group to be archived.
Description	This is an optional description to further identify the group.

Location	This is the name of the History Manager for the archive group. Press the **Modify** button next to this field to display a list of History managers.
Mode	ACTIVE or INACTIVE. The archive group must be active in order to archive. The default mode is INACTIVE. You can not change the mode directly in this field. To change the mode, use the **Action** menu.
Basis	This is the method for scheduling archival for the archive group. The Basis can be Timed or Manual. The default is Timed. When the Basis is Timed, you must specify a Start Time and Time Interval as described below. When the Basis is Manual, you start archival for the group using the Archive Log window as described in Section 4.4.3.2, How to Archive Logs.
Start Time	When the Basis is Timed you must enter a start time to specify how far back in time the system must search for data to archive. You should specify a time at least slightly farther back than you actually require to ensure that you archive all data you intend to. Start Time is entered using the following format: day of the month, month (first three letters), year (last two digits), hours:minutes:seconds. For example: `07 Jun 94 09:35:00`. A last archived time is kept for each log. When data for a log is archived, data is taken from the last archive time to present. If a value in a log is modified, the last archived time is set to the time of the modified entry. This ensures that the new value is archived.
Time Interval	When the Basis is Timed you must enter a time interval to specify how often the archive group is archived. The interval is expressed in minutes, hours, days, or weeks. For example 30m = 30 minutes, 8h = 8 hours, 15d = 15 days, and 4w = 4 weeks. Monthly is not a valid time interval. The time interval should be 1/4 to 1/3 of the log time period.
Archive Device	This is the name of the archive device (as defined in the Archive device window) where the data will be archived. Click on the **Modify** button by this field to display a list of available devices.
Log Type	This is the type of log to be archived. All logs assigned to an archive group must be the same log type. The default is **Numeric**. To select another type (Report or Message), click on the target on the right side of this field to display a list of log types.
Logs in Group	This field displays the logs assigned to this archive group. This is done via the Attribute Edit window for individual logs.
Next Archive Time	This time is filled in by the History Builder. It is the next time an archive will be performed on the logs in this group.
Messages	This field displays ERROR and INFO messages that are generated as you use this window.

The menus for the Archive Device window are shown in Figure 3-114.

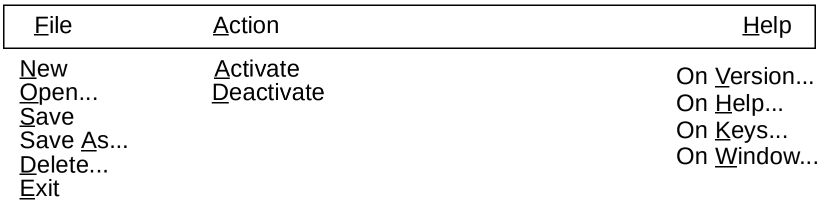


Figure 3-114. Archive Device and archive Group Menu Bar

3.6.10.1 File Menu

See Section 3.6.1, Common AdvaInform History Menus.

3.6.10.2 Action Menu

Action - Activate

Activate the archive device (or group).

Action - Deactivate

Deactivate the archive device (or group).

3.6.10.3 Help Menu

See Section 3.6.1, Common AdvaInform History Menus.

3.6.11 Directory Maintenance Window

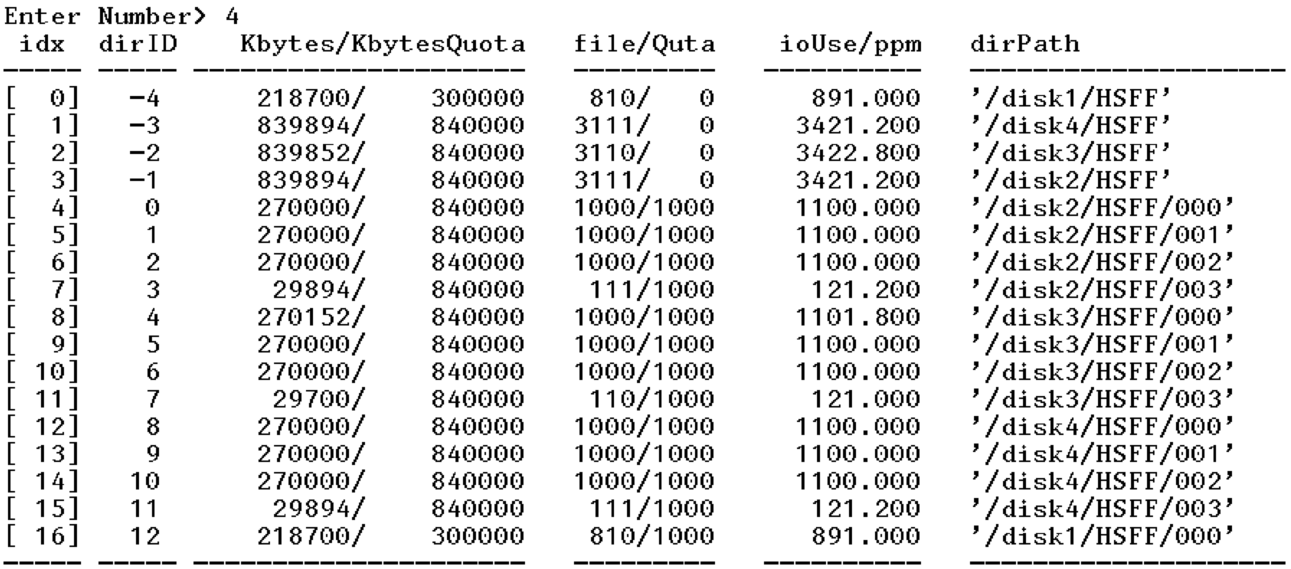
Use the Directory Maintenance window to configure directories to store file-based logs, Figure 3-116. File-based storage is applicable for numeric logs whose Storage Type attribute is configured to be either TYPE1 (default) or TYPE2.

3.6.11.1 Guidelines for Configuring File-based Storage

When History is installed, the disk that History resides on is allocated 200 MB for filed-based storage. You can change this allocation or allocate space on other disks via the Directory Maintenance window.

To allocate space on a disk, you specify the path for the disk, and specify how much memory you want to allocate for files. The system automatically creates directories on the specified disks as needed.

The example directory structure shown in Figure 3-115 illustrates how file-based storage works. This example was produced using the List NumericLog Directories function of hsDBMaint.



```
Enter Number> 4
 idx  dirID    Kbytes/KbytesQuota   file/Quta    ioUse/ppm    dirPath
----- ----- --------------------- ---------   ----------   --------------------
[  0]   -4      218700/    300000   810/   0     891.000   '/disk1/HSFF'
[  1]   -3      839894/    840000  3111/   0    3421.200   '/disk4/HSFF'
[  2]   -2      839852/    840000  3110/   0    3422.800   '/disk3/HSFF'
[  3]   -1      839894/    840000  3111/   0    3421.200   '/disk2/HSFF'
[  4]    0      270000/    840000  1000/1000    1100.000   '/disk2/HSFF/000'
[  5]    1      270000/    840000  1000/1000    1100.000   '/disk2/HSFF/001'
[  6]    2      270000/    840000  1000/1000    1100.000   '/disk2/HSFF/002'
[  7]    3       29894/    840000   111/1000     121.200   '/disk2/HSFF/003'
[  8]    4      270152/    840000  1000/1000    1101.800   '/disk3/HSFF/000'
[  9]    5      270000/    840000  1000/1000    1100.000   '/disk3/HSFF/001'
[ 10]    6      270000/    840000  1000/1000    1100.000   '/disk3/HSFF/002'
[ 11]    7       29700/    840000   110/1000     121.000   '/disk3/HSFF/003'
[ 12]    8      270000/    840000  1000/1000    1100.000   '/disk4/HSFF/000'
[ 13]    9      270000/    840000  1000/1000    1100.000   '/disk4/HSFF/001'
[ 14]   10      270000/    840000  1000/1000    1100.000   '/disk4/HSFF/002'
[ 15]   11       29894/    840000   111/1000     121.200   '/disk4/HSFF/003'
[ 16]   12      218700/    300000   810/1000     891.000   '/disk1/HSFF/000'
----- ----- --------------------- ---------   ----------   --------------------
```

Figure 3-115. Example Directory Structure

When you specify a path for a disk, the directory HSFF is automatically created on the disk. For instance, when the path /disk4 is specified, the directory /disk4/HSFF is created. These HSFF files are assigned a negative dirID number which indicates the order in which their corresponding disks were added. For instance, in Figure 3-115, disk2 is assigned dirID number -1, disk3 has dirID number -2, and so on. The order in which disks are added determines the order they are used.

The HSFF directories have a file quota of 0 (zero). The files are actually stored in subdirectories of HSFF. These subdirectories are automatically created under the HSFF directory as files are stored.

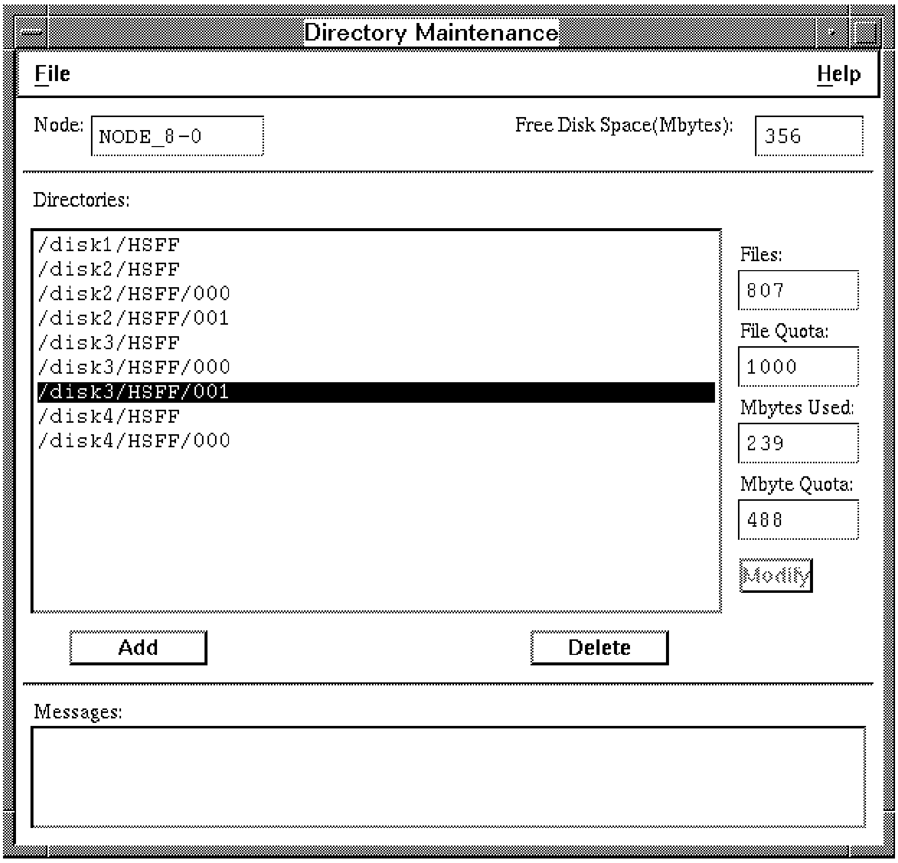
The subdirectories are named using three-digit sequential numbers starting with 000. For instance, /disk4/HSFF/000. These subdirectories have a 1000-file limit (each log constitutes one file). When the number of files exceeds 1000, another subdirectory (/disk4/HSFF/001) is created to store the excess. Subdirectories are added as required until the quota on that disk is reached. New logs create files on the next disk. Disks are used in the order they are added.

In the example above, four disks have been specified for file-based storage: disk1, disk2, disk3, and disk4. The first disk added is disk2 (as indicated by dirID -1). A total of 3111 files are stored on disk in four directories. The directories disk2/HSFF/000, /001, and /002 each hold 1000 files. The directory disk2/HSFF/003 holds 111 and has the capacity for an additional 889 files. The disks disk3 and disk4 have similar patterns. The 003 directories do not contain more files because the total space usage would exceed the quota.

On an Advant Station with AdvaInform software, the disk that History is installed in is disk1. By default, it is designated as the first disk, and 200 MB is allocated for file-based storage. However, in this example, disk1 is the fourth disk in the list, and has 300 MB allocated. This was accomplished by first deleting disk1, and then adding it back to the list.

3.6.11.2 Description of Directory Maintenance Window

To open this window, choose **Directory Maintenance** from the **Tools** menu in the Configuration window.



Node: NODE_8-0

Free Disk Space(Mbytes): 356

Directories:

```
/disk1/HSFF
/disk2/HSFF
/disk2/HSFF/000
/disk2/HSFF/001
/disk3/HSFF
/disk3/HSFF/000
/disk3/HSFF/001
/disk4/HSFF
/disk4/HSFF/000
```

Files: 807

File Quota: 1000

Mbytes Used: 239

Mbyte Quota: 488

Messages:

Figure 3-116. Directory Maintenance Window

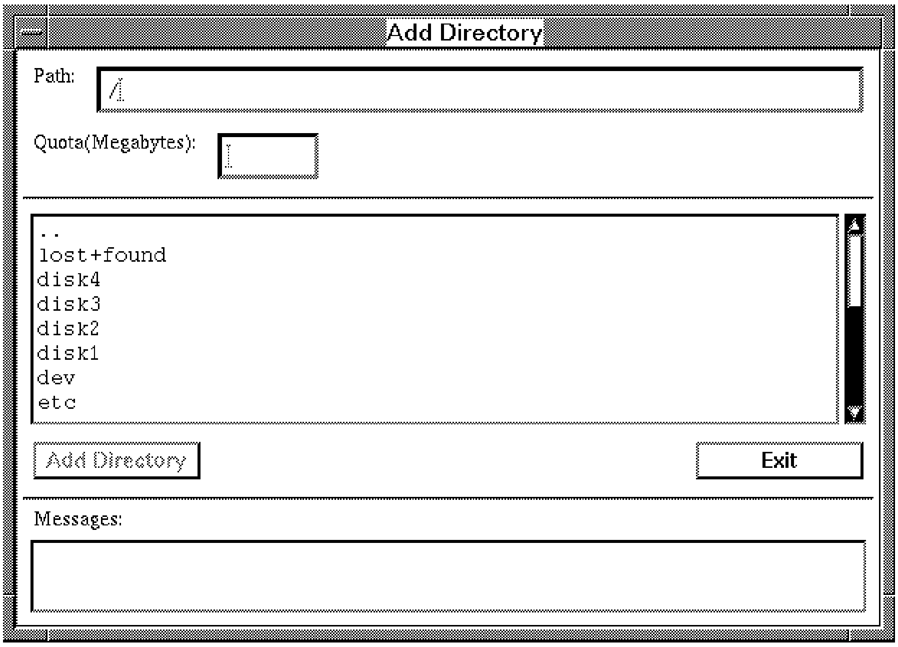
The operation of the widgets for this are described below:

Node	This field indicates the current History Manager.
Free Disk Space	Indicates available space on the selected disk.
Directories	This is a list of directories and sub-directories that have been added for file-based storage. For instance, in Figure 3-116, /disk3/HSFF has two sub-directories 000 and 001. When you select a directory from this list, fields to the right of this list display the data for the selected directory.
Files	This indicates the number of files (logs) in the selected directory. The maximum number of files allowed in a directory is 1000.

File Quota	This indicates the number of files allowed in the directory based on the Mbytes quota you assigned to the directory when you added it. The File Quota can not exceed 1000 regardless of the Mbytes quota you assign it.
Mbytes Used	This indicates the amount of disk space currently being used.
Mbytes Quota	This indicates the amount of memory allocated to the selected directory.
Modify	You can modify the Mbytes quota for any directory directly under the root directory, but not any of its sub-directories. To modify the Mbytes quota, select the directory and then click on **Modify**. This displays a dialog for modifying the Mbytes quota.
Add	Use this button to add a directory for storing file-based logs. See Section 3.6.11.3, Add Button.
Delete	Use this button to remove a directory from the Directory Maintenance window so the directory is no longer available for file-based History data storage. See Section 3.6.11.4, Delete Button.
Messages	This field displays ERROR and INFO messages that are generated as you use this window.

3.6.11.3 Add Button

Use this button to add a directory for file-based storage. When you click **Add**, the Add Directory dialog is displayed, Figure 3-117.



Path: /

Figure 3-117. Add Directory Dialog

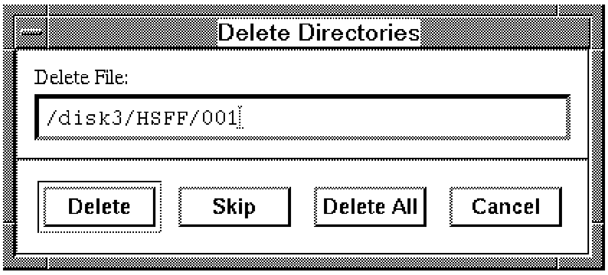
The widgets for this dialog are described below:

Path	This field indicates the path for the disk or directory you are adding. You can enter the path directly in this field, or you can use the selection box below the field. Typically, you simply specify the disk (for instance /disk3). The system will automatically create a directory named HSFF for file storage on the specified disk. The first 1000 files (logs) will be stored in /HSFF/000.
Quota (Megabytes)	Specify the quota for the disk in this field. Enter the number in Megabytes. Typically, the disks you use for History storage are dedicated and not used for any other purpose. Therefore you can generally allocate the full disk capacity for file storage.
Unlabeled Selection Box	This selection box lets you navigate the directory structure of all disks available for History data storage. To navigate down in a directory, double-click on it.
Add Directory	Once you have specified a path to the directory you want to add, click on **Add Directory** to add a selected directory. This button is not active until you select a directory and enter a quota.

Exit	Click on **Exit** to close the Add Directory dialog and return to the Directory Maintenance window.
Messages	This field displays ERROR and INFO messages that are generated as you use this window.

3.6.11.4 Delete Button

Use this button to remove a directory from the list of directories that are available for History data storage. When you click on **Delete**, the Delete Directories dialog is displayed, Figure 3-118.



/disk3/HSFF/001

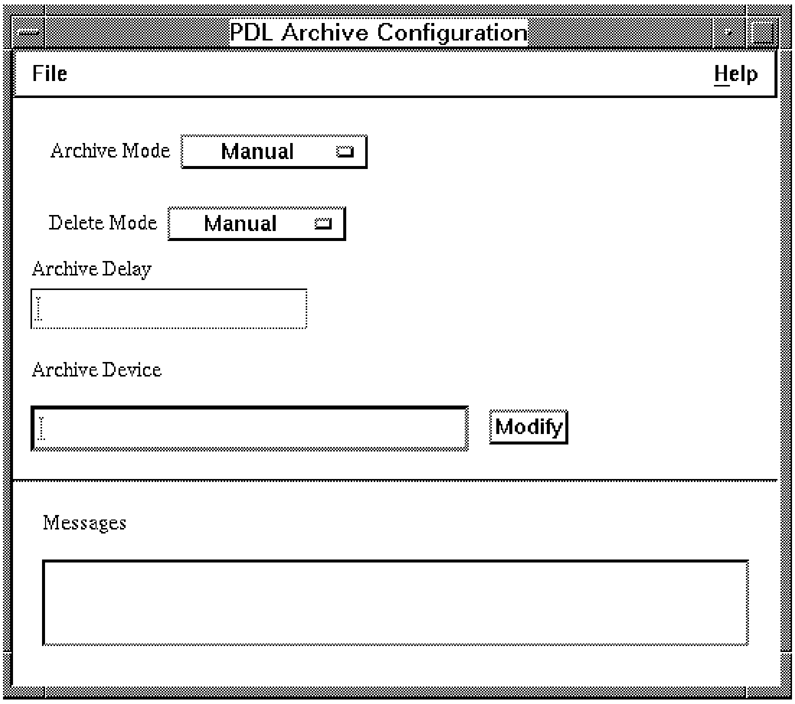
Figure 3-118. Delete Directories Dialog

The operation of the widgets in this dialog are described below:

Delete File	This field indicates the directory selected to delete.
Delete	Click **Delete** to delete the directory indicated in the Delete File field.
Skip	Click **Skip** to skip this directory. When you skip a directory, the next directory in the list is displayed in the Delete File field.
Delete All	Clicking this button will delete all directories listed in the Directory Maintenance window.
Cancel	Clicking this button will cancel the delete function, and return you to the Directory Maintenance window.

3.6.12 PDL Archive Configuration Window (MOD 300)

This window is used to configure PDL archive parameters, Figure 3-119. To open this window, choose **PDL Logs** from the **Tools** menu in the Configuration window, and then choose **Configure PDL Archive** from the pop-up menu.



Archive Mode: Manual

Delete Mode: Manual

Archive Delay

Archive Device

Messages

Figure 3-119. PDL Archive Configuration Window

This window has the following fields and buttons:

Archive Mode	Use this button to select the archive mode. The choices are: Manual - Archive PDLs on demand via the Archive Logs window. On Job End - Archive PDLs automatically after a specified delay from the end of the job. On Batch End - Archive PDLs automatically after a specified delay from the end of the batch.
Delete Mode	Use this button to select a delete mode. The choices are: Manual - Delete PDLs from PDL database on demand via PDL window. After Archive - Delete PDLs from PDL database automatically after they have been archived. After Job End - Delete PDLs after a specified delay from the end of the job. After Batch End - Delete PDLs after a specified delay from the end of the batch.
Archive Delay	You can enter a delay time between the time a job or batch finishes and the time the PDLs are actually archived. This field is only applicable if you select On Job End or On Batch End for Archive Mode. The minimum time is five minutes (5m). This ensures that all data is ready for archival after the end of a job or batch. You can specify a delay of hours (h) or days (d) if you need time to enter additional data that you want to archive.
Archive Device	Enter the archive device in this field. You can enter the device name directly in this field, or use the Modify button to select an archive device.
Messages	This field displays ERROR and INFO messages that are generated as you use this window.

Chapter 4 Runtime Operation

4.1 Product Operation

Historical database configuration and historical data processing are supported by AdvaInform History. Presentation of historical data via trend displays and so on is supported by AdvaCommand software, AdvaSoft for Windows, AdvaInform Display, and Multibus-based trend displays (MOD 300 only).

AdvaInform History automatically performs the collection and storage of data to disk according to the history database configuration. It can transfer data to removable media for long term storage if necessary. Other applications can retrieve data for presentation, calculation, and analysis of data. These include third party packages as well as Advant OCS software.

4.2 Operating Overview

The following runtime functions are described in this section:

- History Status function lets you monitor history log status, and activate/deactivate logs on a log list basis. See Section 4.4.1, History Status for details.
- Start and stop data collection for logs by any one of the following methods:
  - via the History Status function (Section 4.4.1, History Status)
  - on a Log Set basis (Section 4.4.2.1, Activating and De-activating a Log Set) or Log Group Basis (Section 4.4.2.2, Activating and De-activating a Log Group)

    You configure log sets and log groups via the History Builder as described in Chapter 3, Configuration and Application Building. The same windows you use to build log sets and log groups are used to start and stop data collection for them.
  - individually (Section 4.4.2.3, Activating and De-activating an Individual Log)

    This is done via the **Activate** and **Deactivate** menu items in the **Log** menu of the Composite Log window.
- Archive data to tape for long term storage. See Section 4.4.3, Archiving Data for Logs.
- View and delete production data logs (MOD 300 only). See Section 4.4.4, Handling Production Data Logs (MOD 300).

4.3 Runtime Tutorial

Runtime instructions are covered in Section 4.4, Operating Instructions.

4.4 Operating Instructions

Before you can do the procedures described in this section, you must have access to the History Builder. Refer to Section 3.3, Application Start-up for instructions.

4.4.1 History Status

The History Status function provides a convenient method for monitoring numeric log status, and activating/deactivating numeric logs on a log list basis. Access to history status information and functions is via by the History Log Status main window, Figure 4-1. To open this window:

- From AdvaInform: choose **History** - **History Status** from the **AdvaInform** menu.
- From AdvaCommand: configure an external application for History Log Status window.

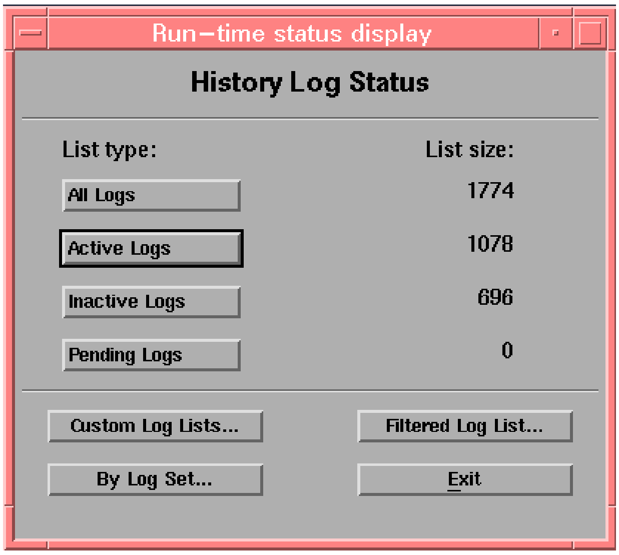


Figure 4-1. History Log Status Main Window

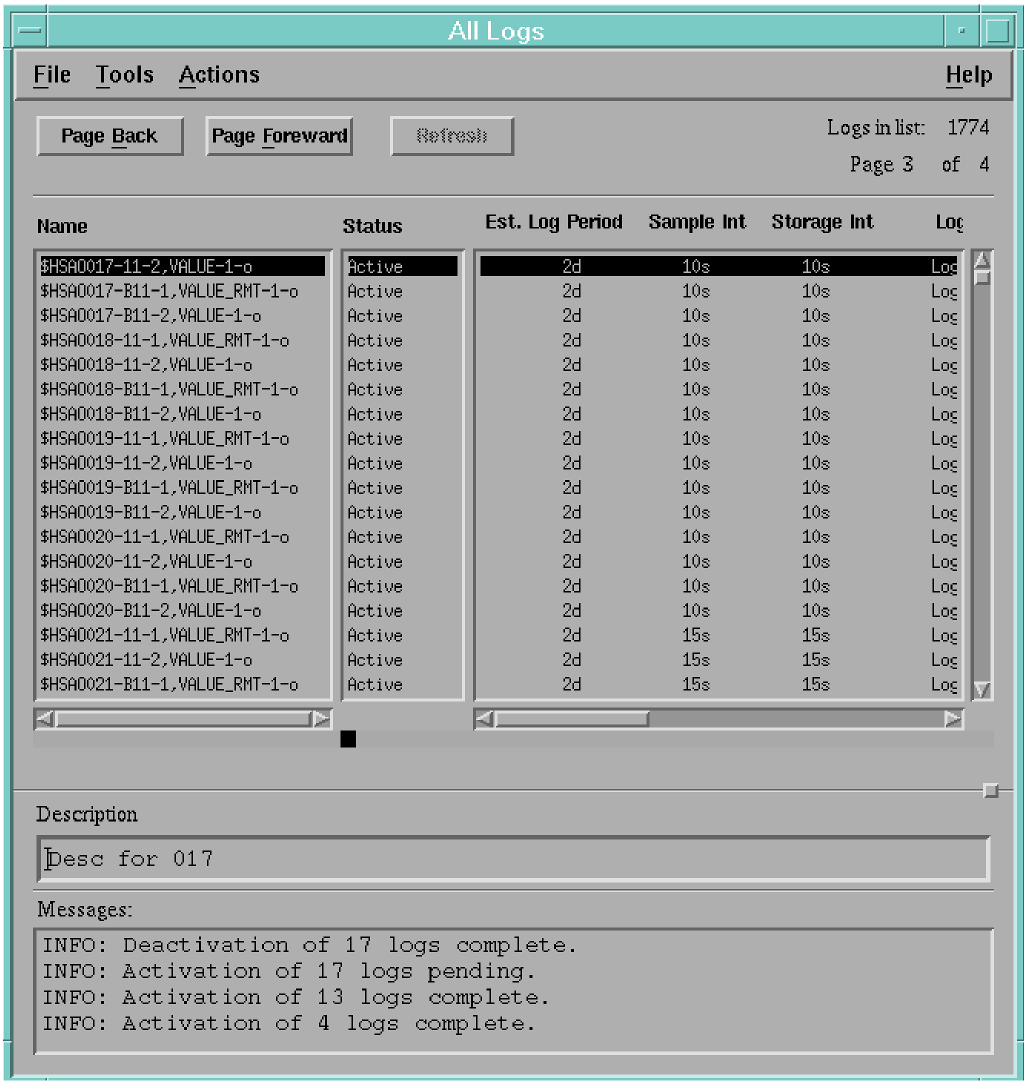
The fields and buttons on this display are described below:

List Size	The List Size fields show the number of logs in the following log lists: All Logs, Active Logs, Inactive Logs, and Pending Logs.
List Type	The List Type buttons open detail windows for the following log lists: All Logs, Active Logs, Inactive Logs, and Pending Logs.
Custom Log Lists	This button lets you open a user-defined (custom) log list for viewing. See Section 4.4.1.3, How to View a Custom List for details. How to build custom log lists is described in Section 4.4.1.2, How to Build a Custom List.
By Log Set...	This button lets you display log status on a log set basis. See Section 4.4.1.4, How to View Detailed History Status Information for a Log Set.
Filtered Log List	This button lets you specify a filter for limiting the logs to be included in the log list. See Section 4.4.1.5, How to Display a Filtered Log List.
Exit	Use this button to exit the Runtime History Status function.

How to display detailed information for a specific type of log list, and how to use the detail display is described in the following sections.

4.4.1.1 How to Display Detailed Runtime History Status Information

To display detailed runtime History status information for a specific type of log list, click on the appropriate button on the History Log Status window. For example, if you click on the **All Logs** button, the detail window for all logs is opened, Figure 4-2.



Logs in list: 1774
Page 3 of 4

Name	Status	Est. Log Period	Sample Int	Storage Int	Log
$HSA0017-11-2,VALUE-1-o	Active	2d	10s	10s	Log
$HSA0017-B11-1,VALUE_RMT-1-o	Active	2d	10s	10s	Log
$HSA0017-B11-2,VALUE-1-o	Active	2d	10s	10s	Log
$HSA0018-11-1,VALUE_RMT-1-o	Active	2d	10s	10s	Log
$HSA0018-11-2,VALUE-1-o	Active	2d	10s	10s	Log
$HSA0018-B11-1,VALUE_RMT-1-o	Active	2d	10s	10s	Log
$HSA0018-B11-2,VALUE-1-o	Active	2d	10s	10s	Log
$HSA0019-11-1,VALUE_RMT-1-o	Active	2d	10s	10s	Log
$HSA0019-11-2,VALUE-1-o	Active	2d	10s	10s	Log
$HSA0019-B11-1,VALUE_RMT-1-o	Active	2d	10s	10s	Log
$HSA0019-B11-2,VALUE-1-o	Active	2d	10s	10s	Log
$HSA0020-11-1,VALUE_RMT-1-o	Active	2d	10s	10s	Log
$HSA0020-11-2,VALUE-1-o	Active	2d	10s	10s	Log
$HSA0020-B11-1,VALUE_RMT-1-o	Active	2d	10s	10s	Log
$HSA0020-B11-2,VALUE-1-o	Active	2d	10s	10s	Log
$HSA0021-11-1,VALUE_RMT-1-o	Active	2d	15s	15s	Log
$HSA0021-11-2,VALUE-1-o	Active	2d	15s	15s	Log
$HSA0021-B11-1,VALUE_RMT-1-o	Active	2d	15s	15s	Log

Description

Desc for 017

Messages:

```
INFO: Deactivation of 17 logs complete.
INFO: Activation of 17 logs pending.
INFO: Activation of 13 logs complete.
INFO: Activation of 4 logs complete.
```

Figure 4-2. History Runtime Status - Detail Window for All Logs

This window has the following fields and buttons:

Name	Displays the log name.
Status	Displays the log status. Status can be: Active Inactive Pending Delete
Attributes	This field contains a column for each attribute that you choose to display via the **Format Attributes List** menu item in the **Tools** menu. See Section 4.4.1.7, How to Change Which Attributes are Displayed in the Detail Window.
Logs in List	This field indicates the total number of logs in the list.
Page *n* of *n*	This field indicates the current page number. You can specify the number of logs per page via the **Configure** menu item in the **Tools** menu. This will affect the total number of pages for the list.
Description	This field displays a description for the log that you select from the Name column. The description is specified when you configure the log in the History Builder. If more than one log is selected, no description is displayed.
Messages	This field displays any messages that have been generated for the selected log. You can clear the messages from this field via the **Clear Messages** menu item in the **Tools** menu.
Page Forward, Page Back	These buttons are used to page forward and backward in the log list.
Refresh	This button refreshes the information in the log list.

The log status is updated automatically. Also, when logs are created or deleted, the Runtime Status display indicates that these changes have occurred.

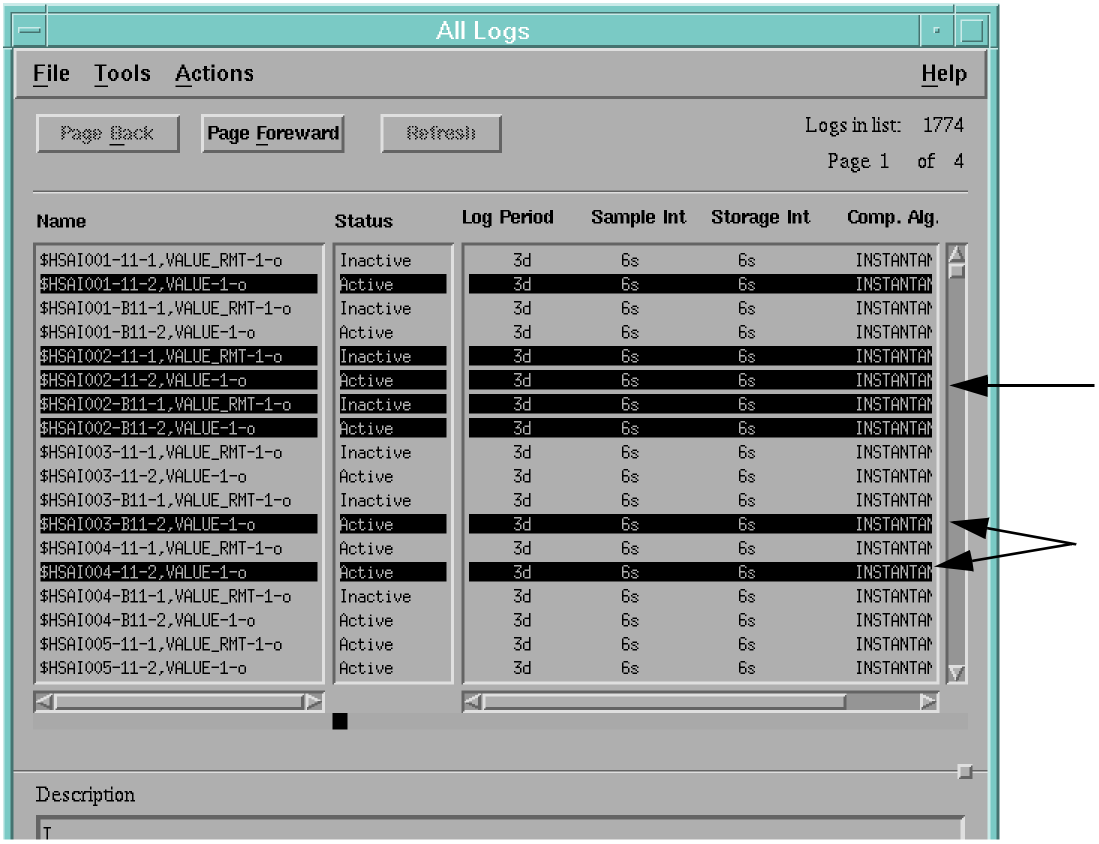
You can display detailed runtime history status information for the following more specific log lists by clicking on the corresponding button:

- active logs
- inactive logs
- pending logs

4.4.1.2 How to Build a Custom List

You can create a custom log list by selecting logs from any one of the standard log lists, and then saving the list to a file. To do this:

1. Open the detail window for a log list (all logs, active logs, inactive logs, or pending logs).
2. Select any combination of logs from the standard list that you want to include in a custom list, Figure 4-3.



Logs in list: 1774
Page 1 of 4

Name	Status	Log Period	Sample Int	Storage Int	Comp. Alg.
$HSAI001-11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN
$HSAI001-11-2,VALUE-1-o	Active	3d	6s	6s	INSTANTAN
$HSAI001-B11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN
$HSAI001-B11-2,VALUE-1-o	Active	3d	6s	6s	INSTANTAN
$HSAI002-11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN
$HSAI002-11-2,VALUE-1-o	Active	3d	6s	6s	INSTANTAN
$HSAI002-B11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN
$HSAI002-B11-2,VALUE-1-o	Active	3d	6s	6s	INSTANTAN
$HSAI003-11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN
$HSAI003-11-2,VALUE-1-o	Active	3d	6s	6s	INSTANTAN
$HSAI003-B11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN
$HSAI003-B11-2,VALUE-1-o	Active	3d	6s	6s	INSTANTAN
$HSAI004-11-1,VALUE_RMT-1-o	Active	3d	6s	6s	INSTANTAN
$HSAI004-11-2,VALUE-1-o	Active	3d	6s	6s	INSTANTAN
$HSAI004-B11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN
$HSAI004-B11-2,VALUE-1-o	Active	3d	6s	6s	INSTANTAN
$HSAI005-11-1,VALUE_RMT-1-o	Active	3d	6s	6s	INSTANTAN
$HSAI005-11-2,VALUE-1-o	Active	3d	6s	6s	INSTANTAN

Shift key for
Contiguous selection

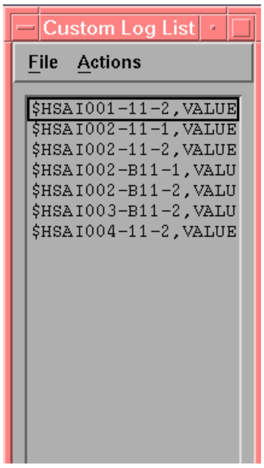
Ctrl key for
non-contiguous
selection

Figure 4-3. Selecting Logs for a Custom List

To select individual non-contiguous logs, press and hold the **Ctrl** key as you select the logs. To select a group of contiguous logs, select the first log, then press and hold the **Shift** key as you select the last log in the contiguous group. You can also drag the mouse to select a contiguous group. To un-select an individual log, press and hold the **Ctrl** key as you click on the selected the log.

3. When you have selected all the logs that you want to include in the custom list, choose **Add to Custom List** from the **Actions** menu.

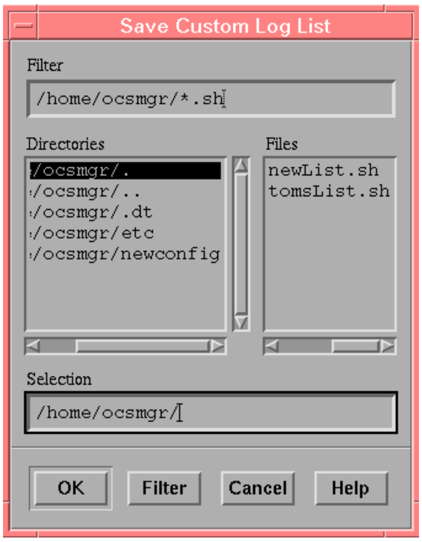
This opens the Custom Log List window, Figure 4-4 which shows the logs that you selected to include in the custom list.



```
$HSAI001-11-2,VALUE
$HSAI002-11-1,VALUE
$HSAI002-11-2,VALUE
$HSAI002-B11-1,VALU
$HSAI002-B11-2,VALU
$HSAI003-B11-2,VALU
$HSAI004-11-2,VALUE
```

Figure 4-4. Custom Log List Dialog

4. Repeat steps 1 through 3 to add logs from other lists to this custom list.
5. If you need to make further changes to the list at this time you can use the **Delete** or **Delete All** menu items in the **Actions** menu; otherwise, choose **Save Custom File** from the **File** menu. This displays a dialog for specifying the file for the new list, Figure 4-5.



Filter: /home/ocsmgr/*.sh

Directories:
```
/ocsmgr/.
/ocsmgr/..
/ocsmgr/.dt
/ocsmgr/etc
/ocsmgr/newconfig
```

Files:
```
newList.sh
tomsList.sh
```

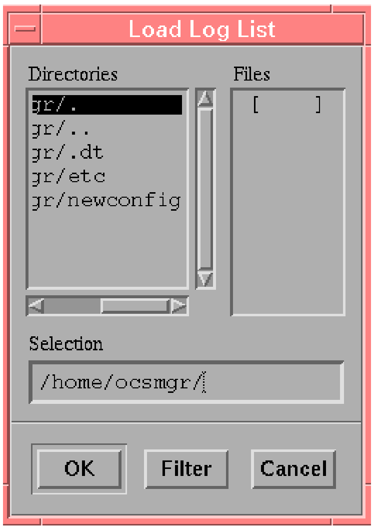
Selection: /home/ocsmgr/

Figure 4-5. Save Custom Log List Dialog

6. Navigate to the appropriate directory in the Directories selection box, enter the file name in the Files field, then click **OK**. This returns you to the Custom Log List dialog.
7. Choose **Close** from the **File** menu to close this dialog.

4.4.1.3 How to View a Custom List

Once you have built one or more custom lists, you can view them via the **Custom Log Lists** button on the History Log Status window. This button displays a dialog for selecting a file that contains a custom log list, Figure 4-6. Use the Directories selection box to navigate to the appropriate directory, and then select the file in the Files selection box. An example custom list is shown in Figure 4-7.

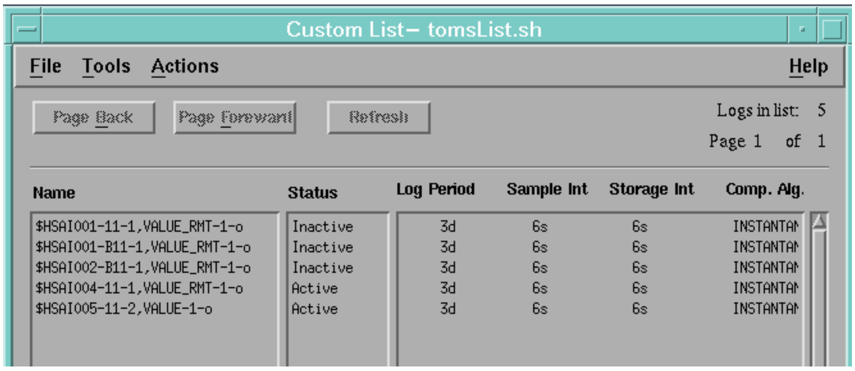


gr/.
gr/..
gr/.dt
gr/etc
gr/newconfig

[ ]

Selection: /home/ocsmgr/

Figure 4-6. Load Log List Dialog



Logs in list: 5
Page 1 of 1

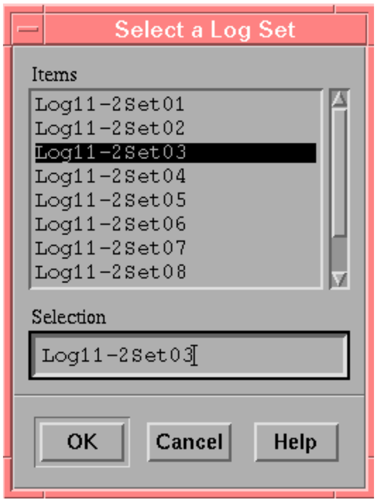
Name	Status	Log Period	Sample Int	Storage Int	Comp. Alg.
$HSAI001-11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN
$HSAI001-B11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN
$HSAI002-B11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN
$HSAI004-11-1,VALUE_RMT-1-o	Active	3d	6s	6s	INSTANTAN
$HSAI005-11-2,VALUE-1-o	Active	3d	6s	6s	INSTANTAN

Figure 4-7. Example Custom Log List

This function is also available via the **Open Custom List** menu item in the **Tools** menu.

4.4.1.4 How to View Detailed History Status Information for a Log Set

To view detailed history status information for a log set, first click on the **By Log Set** button on the History Log Status window. This displays a dialog for selecting the log set whose detailed status information you want to view, Figure 4-8.



Log11-2Set01
Log11-2Set02
Log11-2Set03
Log11-2Set04
Log11-2Set05
Log11-2Set06
Log11-2Set07
Log11-2Set08

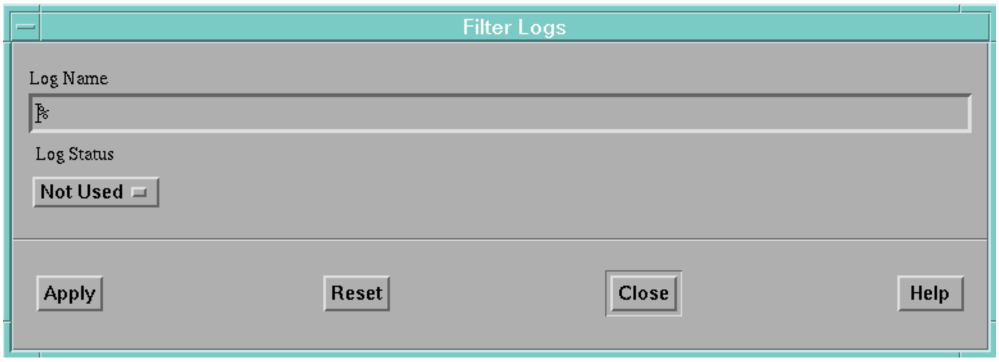
Selection: Log11-2Set03

Figure 4-8. By Log Sets Dialog

Select a log set from the dialog, and then click **OK**.

4.4.1.5 How to Display a Filtered Log List

You can specify a filter to show a specific class of the logs. First click on the **Filtered Log List** button on the History Log Status window to display the Filter Logs dialog, Figure 4-9.



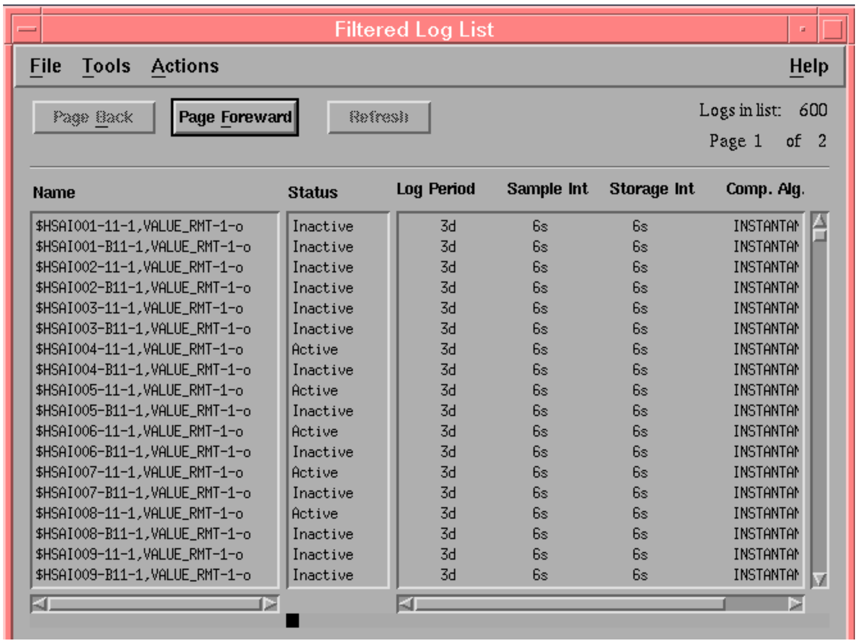
Log Name: %

Log Status: Not Used

Figure 4-9. Filter Logs Dialog

Enter the filter in the Log Name field, and then click **Apply**. The following wildcard characters can be used in the filter: * = any string, % = any single character. You can also use Log Status as an additional filter.

An example of a filtered list is shown in Figure 4-10.



Logs in list: 600
Page 1 of 2

Name	Status	Log Period	Sample Int	Storage Int	Comp. Alg.
$HSAI001-11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN
$HSAI001-B11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN
$HSAI002-11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN
$HSAI002-B11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN
$HSAI003-11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN
$HSAI003-B11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN
$HSAI004-11-1,VALUE_RMT-1-o	Active	3d	6s	6s	INSTANTAN
$HSAI004-B11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN
$HSAI005-11-1,VALUE_RMT-1-o	Active	3d	6s	6s	INSTANTAN
$HSAI005-B11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN
$HSAI006-11-1,VALUE_RMT-1-o	Active	3d	6s	6s	INSTANTAN
$HSAI006-B11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN
$HSAI007-11-1,VALUE_RMT-1-o	Active	3d	6s	6s	INSTANTAN
$HSAI007-B11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN
$HSAI008-11-1,VALUE_RMT-1-o	Active	3d	6s	6s	INSTANTAN
$HSAI008-B11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN
$HSAI009-11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN
$HSAI009-B11-1,VALUE_RMT-1-o	Inactive	3d	6s	6s	INSTANTAN

Figure 4-10. Filtered Log List

In this case, the filter entered in the Log name field Filter Logs dialog is: *_RMT%.

4.4.1.6 How to Activate and Deactivate Logs via the Detail Window

Once you have opened a log list as described in one of the preceding sections, you can activate and deactivate individual logs, a subset of selected logs, or the entire log list. These functions are provided by menu items in the **Actions** menu.

To activate all logs in the current list, choose **Activate All** from the **Actions** menu.

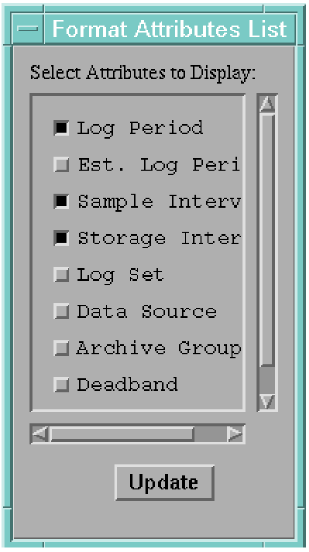
To deactivate all logs in the current list, choose **Deactivate** All from the **Actions** menu.

To activate or deacivate one or more logs in the current log list, first select the logs and then choose **Activate** or **Deactivate** from the **Actions** menu. To select individual non-contiguous logs, press and hold the **Ctrl** key as you select the logs. To select a group of contiguous logs, select the first log, then press and hold the **Shift** key as you select the last log in the contiguous group. To un-select an individual log, press and hold the **Ctrl** key as you click on the selected the log.

4.4.1.7 How to Change Which Attributes are Displayed in the Detail Window

You can show and hide log attributes in the detail window on an individual basis. To do this:

1. Choose **Format Attribute List** from the **Tools** menu. This displays the Format Attribute List dialog, Figure 4-11.



■ Log Period
☐ Est. Log Peri
■ Sample Interv
■ Storage Inter
☐ Log Set
☐ Data Source
☐ Archive Group
☐ Deadband

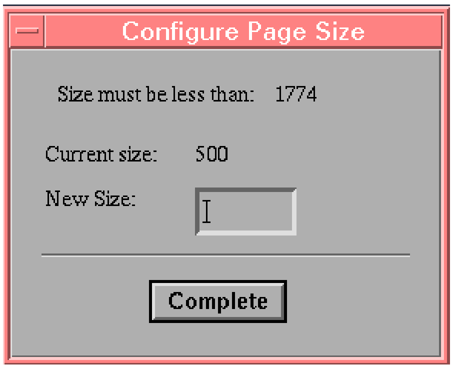
Figure 4-11. Format Attributes List Dialog

2. Select the attributes that you want to show, and deselect the attributes that you want to hide, then click **Update**.

4.4.1.8 How to Configure Page Size

You can specify the number of logs per page. To do this:

1. Choose **Configure** from the **Tools** menu. This displays the Configure Page Size dialog, Figure 4-12.



Size must be less than: 1774

Current size: 500

New Size:

Figure 4-12. Configure Page Size Dialog

This dialog indicates the current number of logs. Page size must be no greater than the current number of logs.

2. Enter the new page size in the New Size field and then click **Complete**. This changes the number of logs per page and number of pages accordingly.

4.4.2 Starting and Stopping Data Collection

You start and stop data collection by activating or deactivating logs. There are four methods:

- Logs can be activated on a list basis via the History Status function. See Section 4.4.1, History Status.
- All logs of a log set can be activated by the Log Set's **Action** menu. See Section 4.4.2.1, Activating and De-activating a Log Set.
- All logs of a log group can be activated by the Log Group's **Action** menu. See Section 4.4.2.2, Activating and De-activating a Log Group.
- A log can be activated individually via the **Activate** and **Deactivate** menu items in the **Log** menu of the Composite Log window. See Section 4.4.2.3, Activating and De-activating an Individual Log.

4.4.2.1 Activating and De-activating a Log Set

A log set is a means to activate and de-activate a number of logs simultaneously. The logs in a log set are assigned to that set via the Composite Log window. The logs must be in the same node, but they can have different log types. To activate or deactivate the logs in a log set:

1. To open the Log Set window, choose **Log Set** from the **Tools** menu in the History Configuration Window, Figure 4-13.

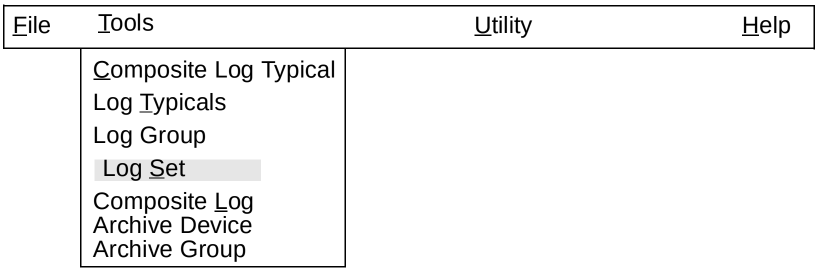


Figure 4-13. Opening the Log Set Window

This displays the Log Set window, Figure 4-14

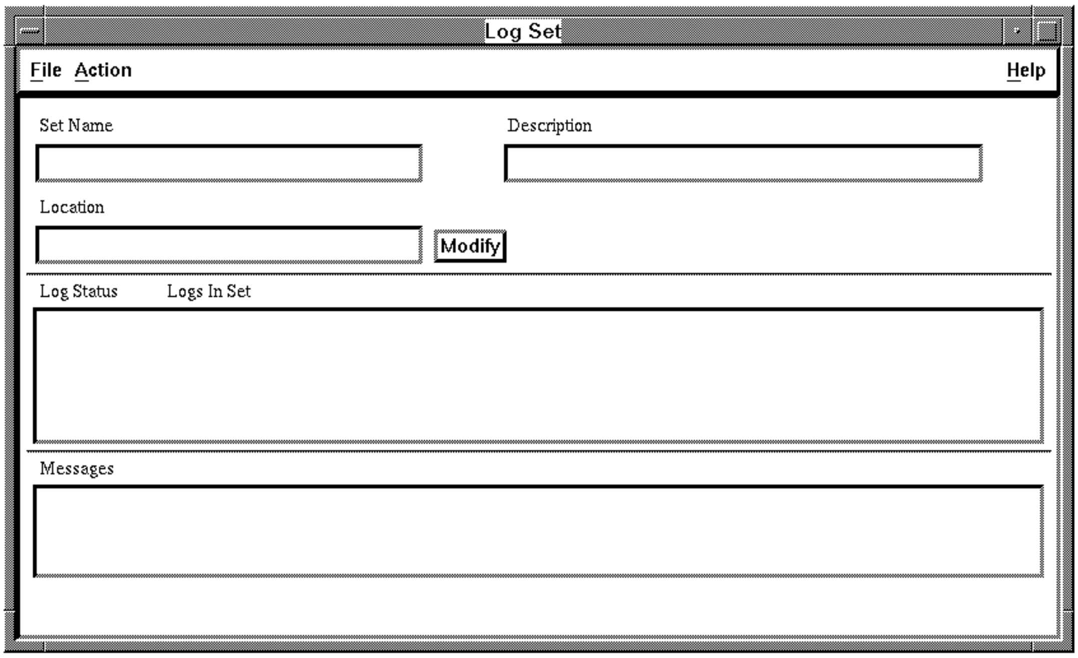
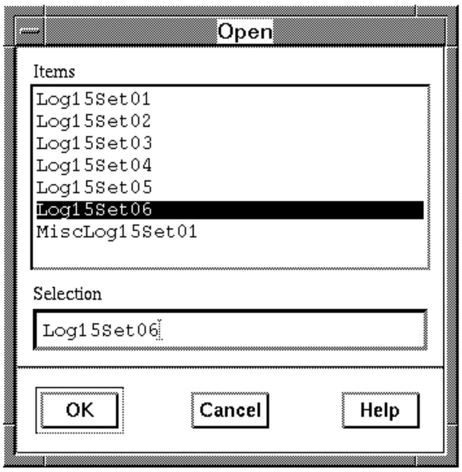


Figure 4-14. Log Set Window

2. Open an existing log set as follows:
    a. Choose **Open** from the **File** menu.

This displays a list of log sets, Figure 4-15, (configured as described in Section 3.5.3, Log Sets and Section 3.5.3.1, How to Assign Logs to a Log Set).



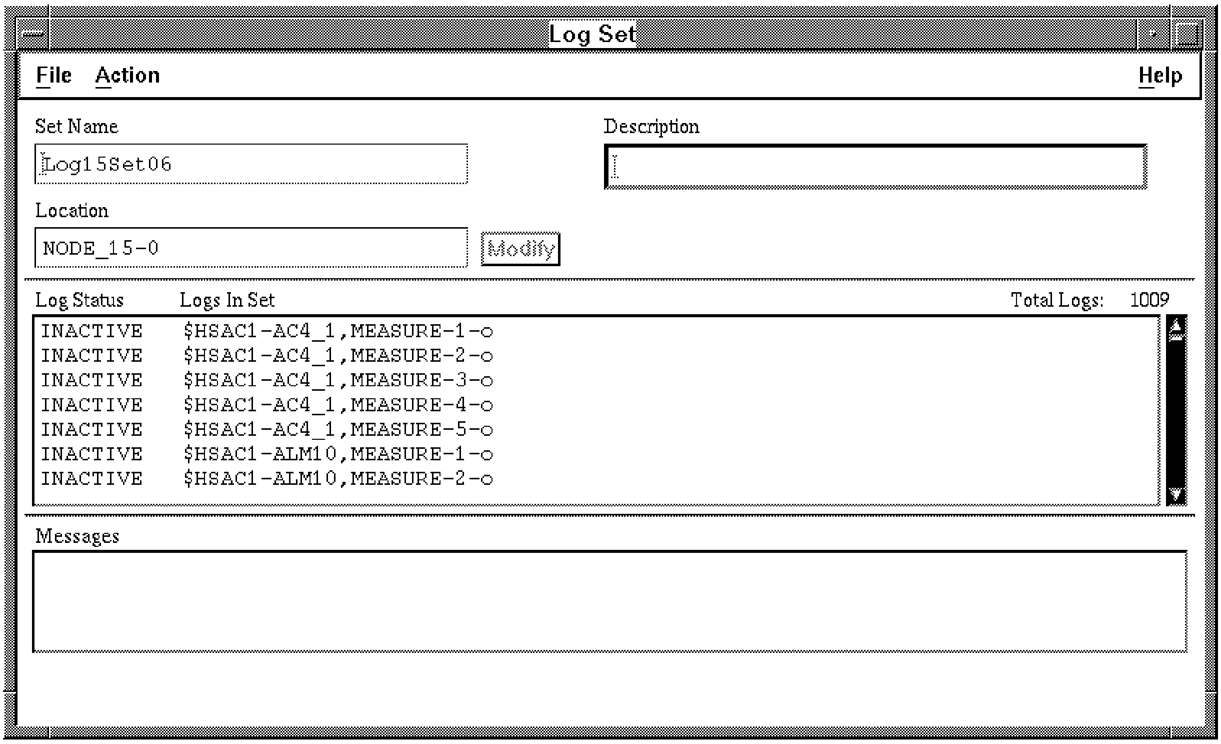
Log15Set01
Log15Set02
Log15Set03
Log15Set04
Log15Set05
Log15Set06
MiscLog15Set01

Selection: Log15Set06

Figure 4-15. Open Log Set Dialog Box

b. Select a log set (for example LogSet06) and then click **OK**.

This displays the Log Set data in the window, Figure 4-16.



Set Name: Log15Set06

Description:

Location: NODE_15-0

Total Logs: 1009

```
Log Status    Logs In Set
INACTIVE      $HSAC1-AC4_1,MEASURE-1-o
INACTIVE      $HSAC1-AC4_1,MEASURE-2-o
INACTIVE      $HSAC1-AC4_1,MEASURE-3-o
INACTIVE      $HSAC1-AC4_1,MEASURE-4-o
INACTIVE      $HSAC1-AC4_1,MEASURE-5-o
INACTIVE      $HSAC1-ALM10,MEASURE-1-o
INACTIVE      $HSAC1-ALM10,MEASURE-2-o
```

Figure 4-16. Log Set Display Window

c. To start data collection by all logs in the log set, choose **Activate** from the **Action** menu, Figure 4-17.

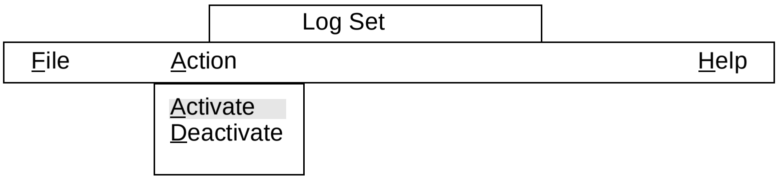
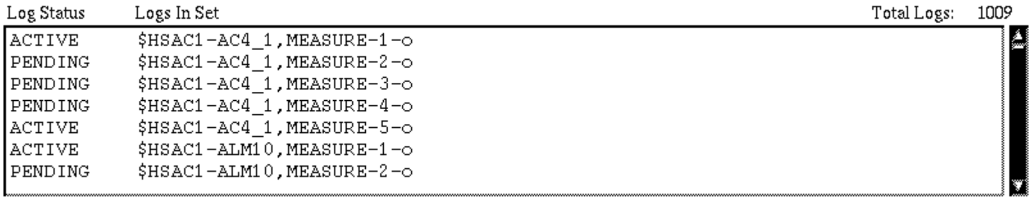


Figure 4-17. Activating a Log Set

This displays a message that indicates activation of the log set is started:

```
INFO: Activation of ‘Log15Set06’ Started.
```

The Log Set window is not dynamically updated. If you want to verify that the logs in the log set are active, re-open the window, Figure 4-18.



Total Logs: 1009

Log Status	Logs In Set
ACTIVE	$HSAC1-AC4_1,MEASURE-1-o
PENDING	$HSAC1-AC4_1,MEASURE-2-o
PENDING	$HSAC1-AC4_1,MEASURE-3-o
PENDING	$HSAC1-AC4_1,MEASURE-4-o
ACTIVE	$HSAC1-AC4_1,MEASURE-5-o
ACTIVE	$HSAC1-ALM10,MEASURE-1-o
PENDING	$HSAC1-ALM10,MEASURE-2-o

Figure 4-18. verification that Logs in Log Set are Active

d. To deactivate an activated log set, choose **Deactivate** from the **Action** menu.

4.4.2.2 Activating and De-activating a Log Group

The main function of a log group is to configure a number of logs that have the same data storage parameters. The logs of a log group must be in the same node and be the same log type. During runtime, a log group provides a means to activate and de-activate logs simultaneously. To activate or deactivate a log group:

1. To open the Log Group window choose **Log Group** from the **Tools** menu in the History Configuration Window.

   This displays the Log Group window, Figure 4-19.

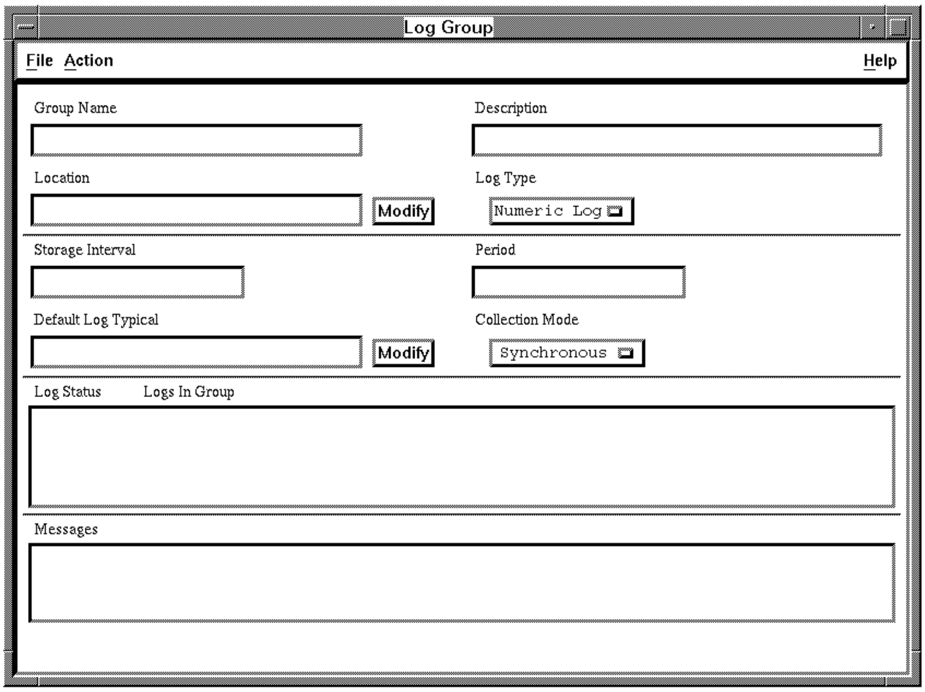
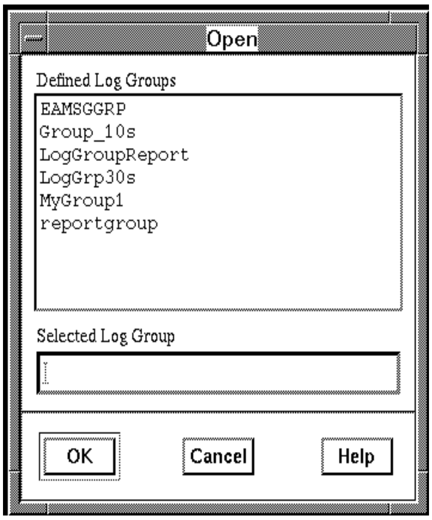


Figure 4-19. Log Group Window

2. Open an existing log group as follows:
   a. Choose **Open** from the **File** menu.

      This displays a list of log groups, Figure 4-20.

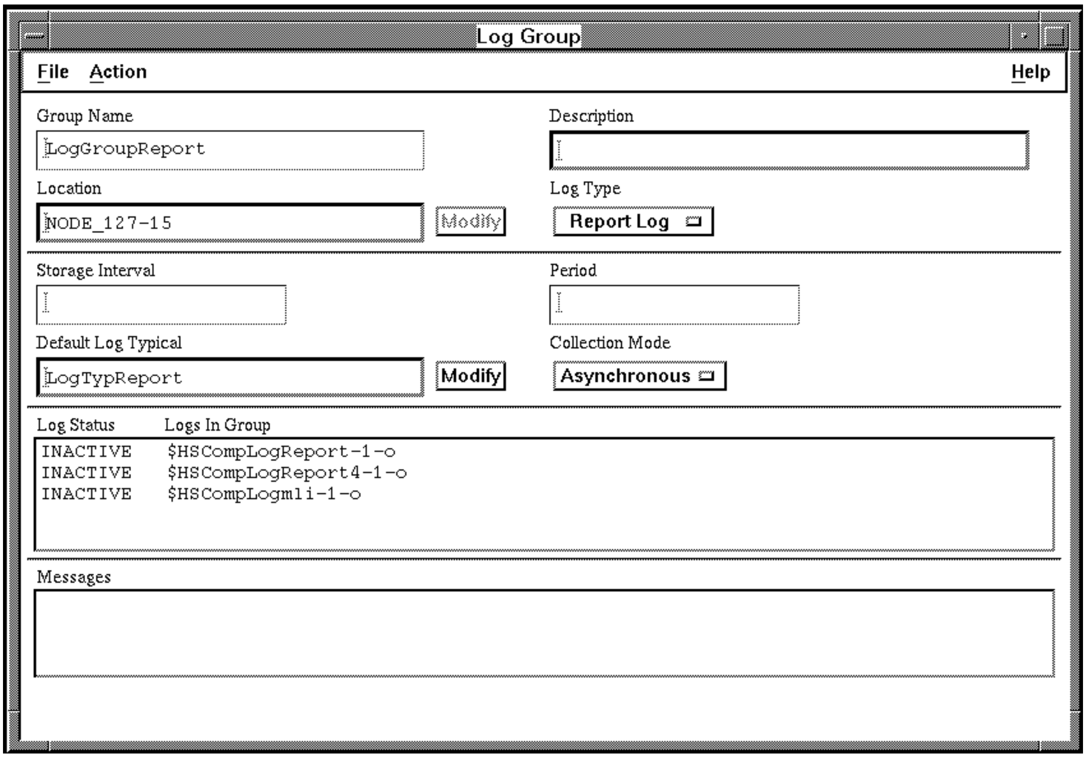


EAMSGGRP
Group_10s
LogGroupReport
LogGrp30s
MyGroup1
reportgroup

Figure 4-20. Open Log Groups Dialog Box

b. Select an existing log group and then click **OK**.

This displays the log group data in the window, Figure 4-21.



Group Name: LogGroupReport
Description:
Location: NODE_127-15
Log Type: Report Log
Storage Interval:
Period:
Default Log Typical: LogTypReport
Collection Mode: Asynchronous

```
Log Status   Logs In Group
INACTIVE     $HSCompLogReport-1-o
INACTIVE     $HSCompLogReport4-1-o
INACTIVE     $HSCompLogmli-1-o
```

Messages

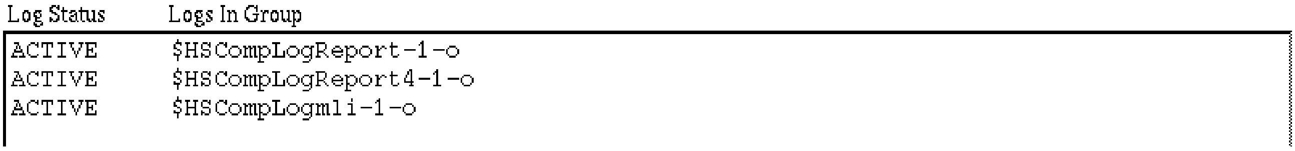
Figure 4-21. Log Group Window w/Data for Selected Log Group

c. To start data collection by all logs in the log group, choose **Activate** from the **Action** menu.

This displays a message that indicates activation of the log group has started:

```
INFO: Activation of 'LogGroupReport' Started.
```

The Log Group window is not dynamically updated. To verify that the logs in the log group are active, re-open the log group, Figure 4-22.



```
Log Status   Logs In Group
ACTIVE       $HSCompLogReport-1-o
ACTIVE       $HSCompLogReport4-1-o
ACTIVE       $HSCompLogmli-1-o
```

Figure 4-22. Log Set Activated Message in Log Group Window

d. To deactivate an active log, choose **Deactivate** from the **Action** menu.

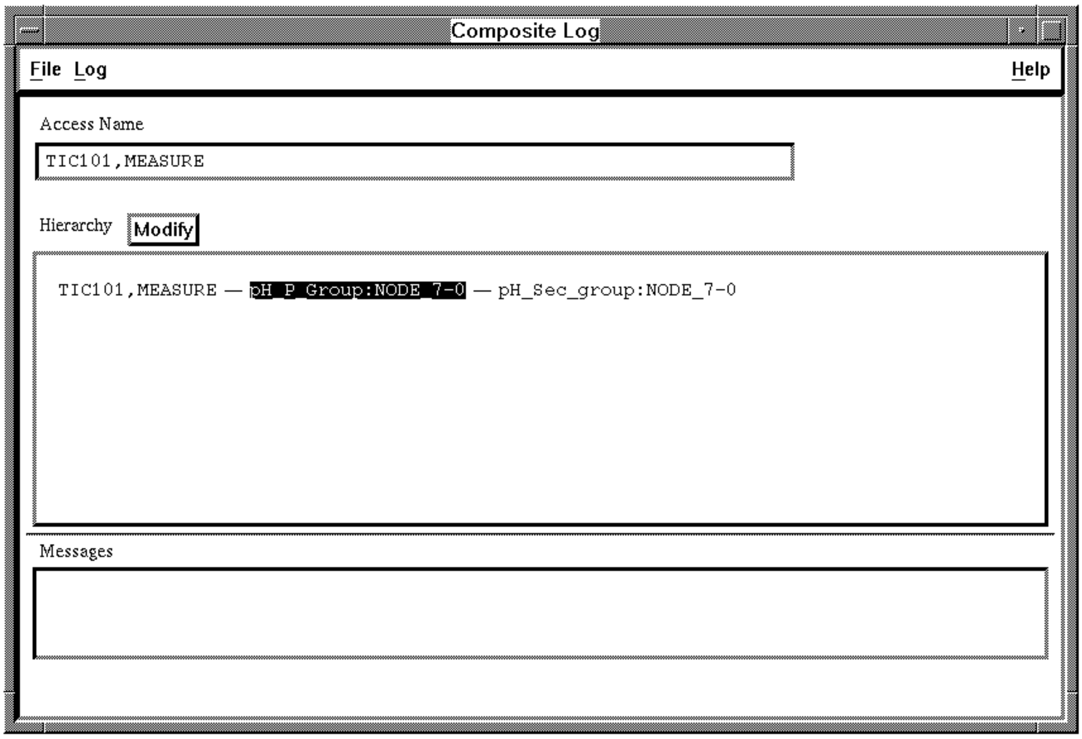
4.4.2.3 Activating and De-activating an Individual Log

A log can be activated or de-activated individually via the Composite Log window.

1. Open the Composite Log Window.
2. To display the log to be activated, choose **Open** from the **File** menu in the Composite Log Window.

   This displays a list of the logs.
3. Select the log you want to activate and click **OK**.

   This displays the selected log in the window.
4. Click on the specific log in the Log Hierarchy box that you want to activate (or deactivate) and then choose **Activate** (or **Deactivate**) from the **Log** menu, Figure 4-23. To select more than one log in the hierarchy to activate (or deactivate), hold down the left mouse button and drag to select them.



TIC101,MEASURE

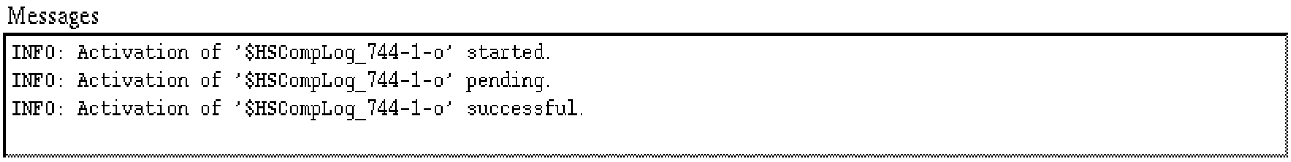
TIC101,MEASURE — pH_P_Group:NODE_7-0 — pH_Sec_group:NODE_7-0

Figure 4-23. Log Selected in Composite Log Window

When you attempt to activate a log, the log may be activated immediately, or activation may be delayed, depending upon how the log's Activation Time attribute is defined. Activation Time determines the earliest allowable time that the log can be activated. In addition, the activation time also determines the hour, minute, and second when the first point is collected.

If Activation Time is disabled, or if the Activation Time has already occurred, the log will be activated immediately. If Activation Time is enabled and specifies some time in the future, the log will not be activated until that time.

A series of messages indicate the status of the activation request. The first message indicates that activation has started. A second message indicates that activation is pending. When the log is actually activated, an Activation Successful message is displayed.

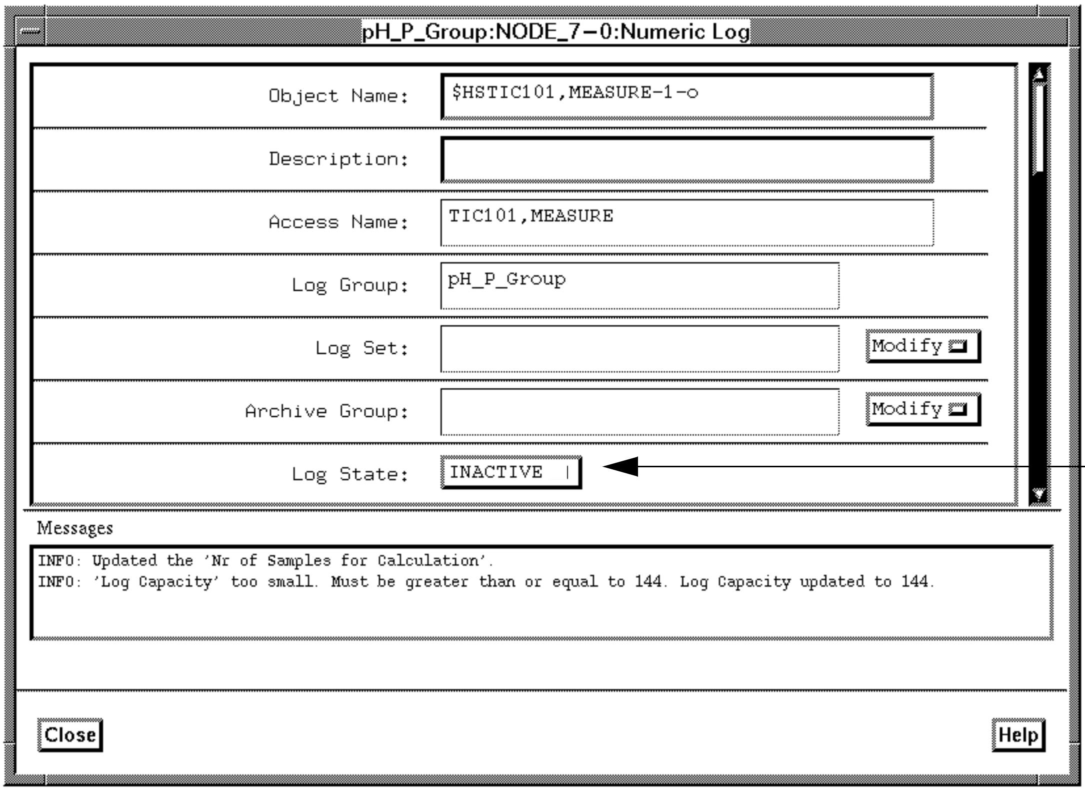


Messages

```
INFO: Activation of '$HSCompLog_744-1-o' started.
INFO: Activation of '$HSCompLog_744-1-o' pending.
INFO: Activation of '$HSCompLog_744-1-o' successful.
```

Figure 4-24. Log activation Messages

You can also see the current log state via the Log Attribute Edit window, Figure 4-25.



pH_P_Group:NODE_7-0:Numeric Log

Object Name: $HSTIC101,MEASURE-1-o

Description:

Access Name: TIC101,MEASURE

Log Group: pH_P_Group

Log Set:

Archive Group:

Log State: INACTIVE

Messages

```
INFO: Updated the 'Nr of Samples for Calculation'.
INFO: 'Log Capacity' too small. Must be greater than or equal to 144. Log Capacity updated to 144.
```

Log State
Attribute

Figure 4-25. Log Attribute Edit Window

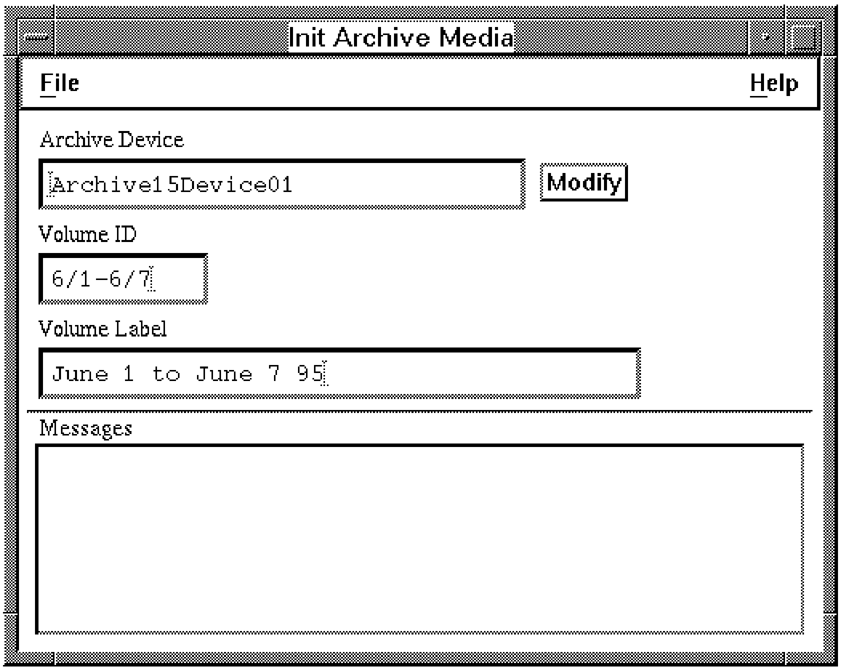
4.4.3 Archiving Data for Logs

This section describes how to use the AdvaInform History archive functions to:

- initialize the archive media. This makes the DAT tape ready to accept archived log data.
- archive (copy) logs from the run time history database to an archive media (DAT).
- view the archive database on the History node internal drive.
- restore archived logs from the archive media to the restored history database.
- delete restored archive logs from the restored history database.
- add archive entries from the archive database on the archive media to the archive database on the History node internal drive.
- delete archive entries from the archive database on the History node internal drive.

4.4.3.1 How to Initialize the Media

You must initialize each new tape that you install in the DAT drive prior to archiving data on it. This is done via the Initialize Media window, Figure 4-26.



Archive Device: Archive15Device01

Volume ID: 6/1-6/7

Volume Label: June 1 to June 7 95

Figure 4-26. Initialize Media Window

This window has the following fields:

Archive Device	This is the name of the archive device where the archive tape (media) is installed (See Section 3.5.7, Configure History Node to Support Archive Function). Click on the **Modify** button by this field to display a list of available devices.
Volume ID	This is a name you assign to this specific tape.
Volume Label	This is an optional descriptor to further identify the tape.
Messages	This field displays messages regarding the activities you perform in this window.

To initialize the archive media:

1. Insert a new tape in the DAT drive.
2. In the AdvaInform History window, choose **Archive** from the **Tools** menu.

   This displays a cascade menu with archive choices.
3. Choose **Initialize Archive Media** from the cascade menu.

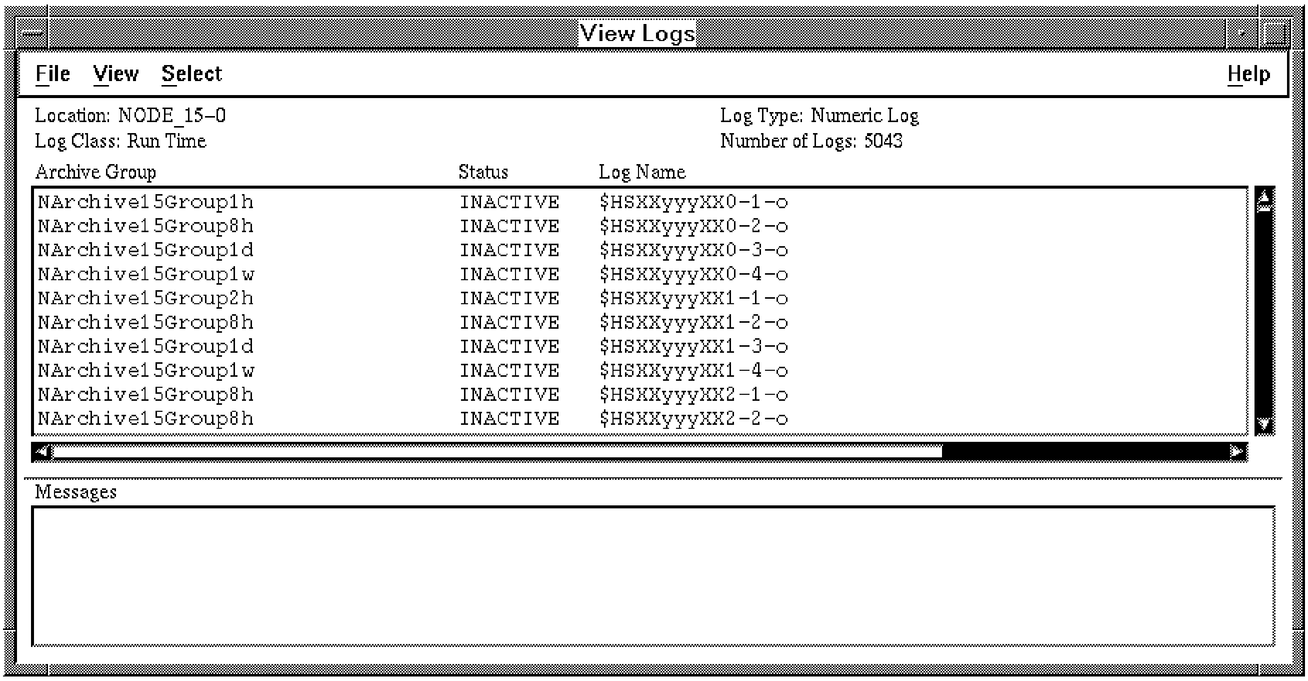
   This opens the Initialize Archive Media window. The fields are blank (un-defined) initially.
4. Enter the name of the archive device where the tape is installed. Click on the **Modify** button by this field to display a list of available devices.
5. Enter a Volume ID.
6. Enter a Volume Label.
7. Choose **Execute** from the **File** menu.

   If you specify the initialize media parameters correctly, a message indicating the initialization was successful will be displayed in the message box. If one or more parameters were defined incorrectly, an error message will be displayed.

4.4.3.2 How to Archive Logs

When you archive logs you copy the logs from the run time portion of the history database to the archive media. This functionality is provided by three different windows (View Logs, View Report Logs, and View Production Data Logs) depending on the kind of log you are archiving.

For numeric, message, and SPC logs, this functionality is supported by the View Logs window Figure 4-27.



Location: NODE_15-0
Log Type: Numeric Log
Log Class: Run Time
Number of Logs: 5043

Archive Group	Status	Log Name
NArchive15Group1h	INACTIVE	$HSXXyyyXX0-1-o
NArchive15Group8h	INACTIVE	$HSXXyyyXX0-2-o
NArchive15Group1d	INACTIVE	$HSXXyyyXX0-3-o
NArchive15Group1w	INACTIVE	$HSXXyyyXX0-4-o
NArchive15Group2h	INACTIVE	$HSXXyyyXX1-1-o
NArchive15Group8h	INACTIVE	$HSXXyyyXX1-2-o
NArchive15Group1d	INACTIVE	$HSXXyyyXX1-3-o
NArchive15Group1w	INACTIVE	$HSXXyyyXX1-4-o
NArchive15Group8h	INACTIVE	$HSXXyyyXX2-1-o
NArchive15Group8h	INACTIVE	$HSXXyyyXX2-2-o

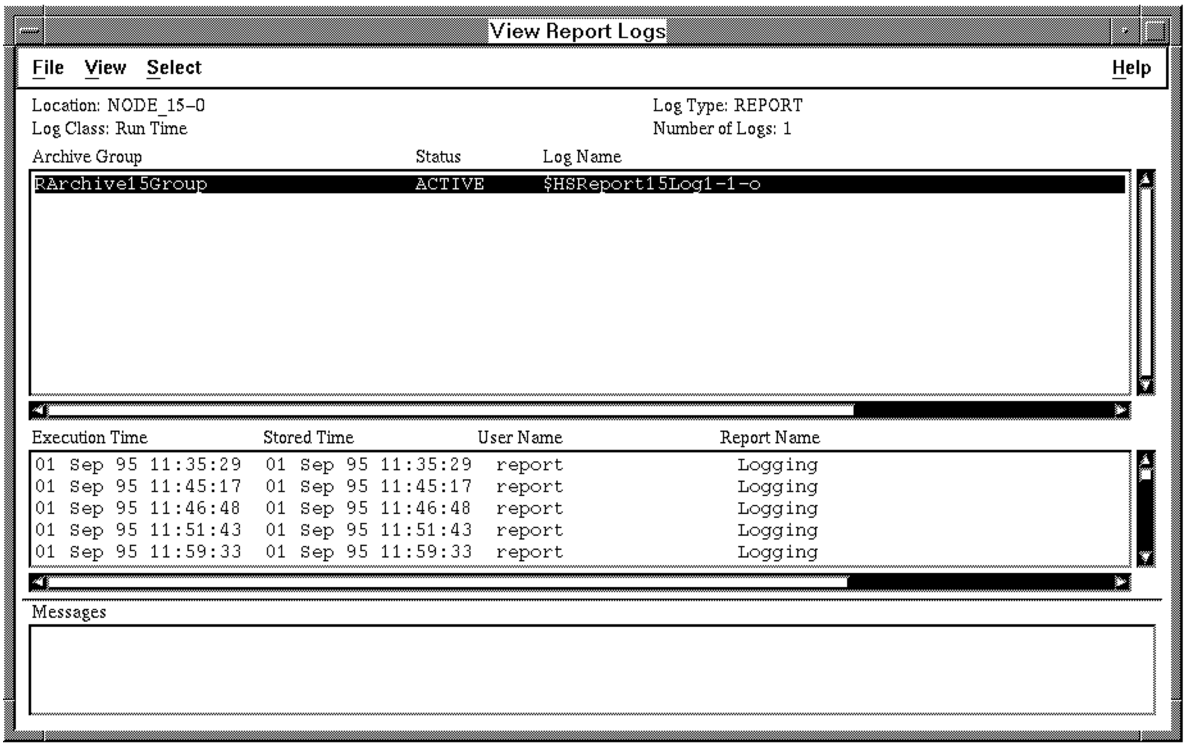
Messages

Figure 4-27. View Logs Window

This window has the following fields:

Location	Indicates the node where the displayed logs reside.
Log Class	Indicates whether the displayed logs are Run Time or Restored.
Log Type	Indicates the type of logs displayed (Numeric, Message, or SPC)
Number of Logs	Indicates the number of logs in the displayed list.
Archive Group	Indicates the archive group that a displayed log is assigned to.
Status	Indicates whether the status of a displayed log is ACTIVE or INACTIVE.
Log name	Indicates the name of the displayed log.
Messages	Displays messages associated with Archive Logs window.

For report logs this functionality is supported by the View Report Logs window, Figure 4-28.



Location: NODE_15-0
Log Class: Run Time
Log Type: REPORT
Number of Logs: 1

Archive Group	Status	Log Name
RArchive15Group	ACTIVE	$HSReport15Log1-1-o

Execution Time	Stored Time	User Name	Report Name
01 Sep 95 11:35:29	01 Sep 95 11:35:29	report	Logging
01 Sep 95 11:45:17	01 Sep 95 11:45:17	report	Logging
01 Sep 95 11:46:48	01 Sep 95 11:46:48	report	Logging
01 Sep 95 11:51:43	01 Sep 95 11:51:43	report	Logging
01 Sep 95 11:59:33	01 Sep 95 11:59:33	report	Logging

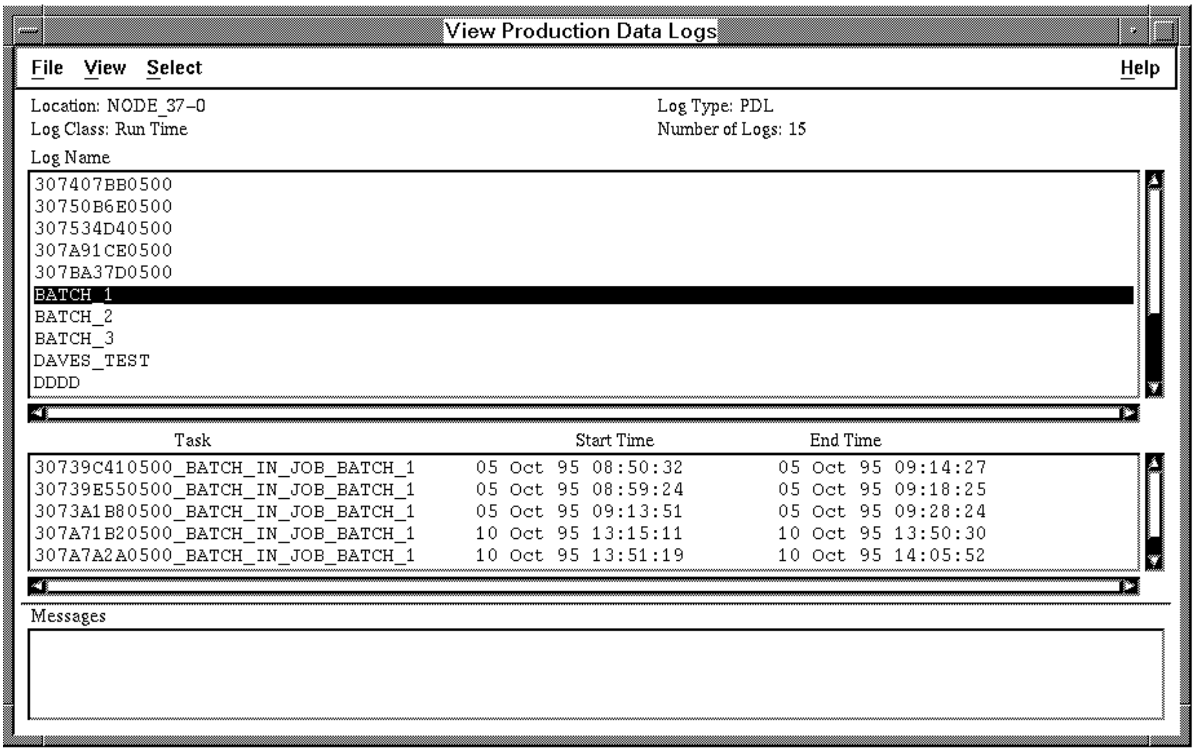
Messages

Figure 4-28. View Report Logs Window

This window has the following fields:

Location	Indicates the node where the displayed logs reside.
Log Class	Indicates whether the displayed logs are Run Time or Restored.
Log Type	Indicates the type of logs displayed REPORT is only valid type.
Number of Logs	Indicates the number of logs in the displayed list.
Archive Group	Indicates the archive group that a displayed log is assigned to.
Status	Indicates the status of a displayed log: ACTIVE or INACTIVE.
Log name	Indicates the name of the displayed log.
Execution Time	Lists the execution times for the reports.
Stored Time	Lists the times when the executed report was stored.
User Name	Lists the user that executed the report.
Report Name	Lists the name of the report that was executed.
Messages	Displays messages for Archive Logs window.

For PDLs (MOD 300) this functionality is supported by the View Production Data Logs window, Figure 4-28.



Location: NODE_37-0
Log Class: Run Time
Log Type: PDL
Number of Logs: 15

```
307407BB0500
30750B6E0500
307534D40500
307A91CE0500
307BA37D0500
BATCH_1
BATCH_2
BATCH_3
DAVES_TEST
DDDD
```

```
30739C410500_BATCH_IN_JOB_BATCH_1    05 Oct 95 08:50:32    05 Oct 95 09:14:27
30739E550500_BATCH_IN_JOB_BATCH_1    05 Oct 95 08:59:24    05 Oct 95 09:18:25
3073A1B80500_BATCH_IN_JOB_BATCH_1    05 Oct 95 09:13:51    05 Oct 95 09:28:24
307A71B20500_BATCH_IN_JOB_BATCH_1    10 Oct 95 13:15:11    10 Oct 95 13:50:30
307A7A2A0500_BATCH_IN_JOB_BATCH_1    10 Oct 95 13:51:19    10 Oct 95 14:05:52
```

Figure 4-29. View Production Data Logs Window

This window has the following fields:

Location	Indicates the node where the displayed logs reside.
Log Class	Indicates whether the displayed logs are Run Time or Restored.
Log Type	Indicates the type of logs displayed. PDL is only valid type.
Number of Logs	Indicates the number of logs in the displayed list.
Log Name	The log name is derived from the name of the root task.
Task	If the PDL is a job created by Batch 300, Task is a list of batches in the job. If the PDL is a batch created by TCL, Task is the same as the log name.
Start Time & End Time	Start Time & End Time specify the time range for the Batch or Job
Messages	Displays messages for View Production Data Logs window.

To archive run time logs:

1. In the AdvaInform History window, choose **Archive** from the **Tools** menu, and then choose **View Logs** (or **View Report Logs** or **View Production Data Logs**) from the cascade menu.

   This opens the View Logs (or View Report Logs or View Production Data Logs) window. Initially, the fields are blank (un-defined).

2. Specify a list or range of logs from which you can select the logs you want to archive.

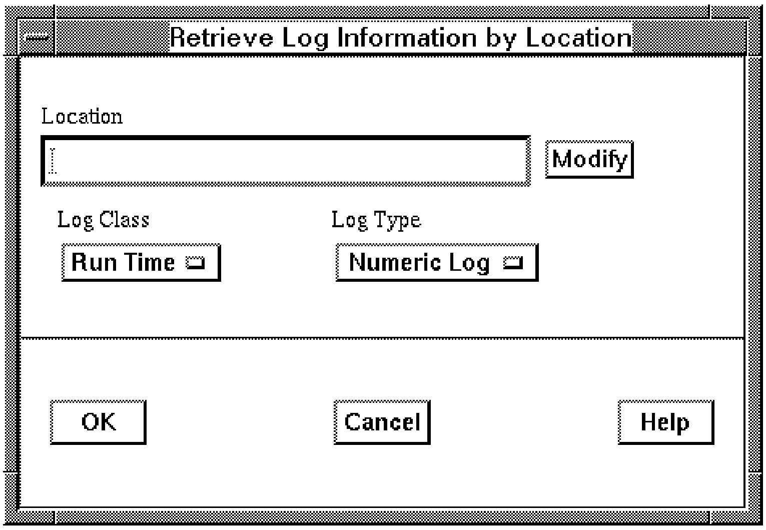
   a. Choose **Retrieve Logs** from the **File** menu.

      This displays a cascade menu where you can choose either **Retrieve by Location** or **Retrieve by Group**.

      When you choose **Retrieve by Location**, the Retrieve Log Information by Location dialog box is displayed, Figure 4-38. This dialog box provides the means to display a list of logs for a specific node (Location), based on Log Class and Log Type.

NOTE

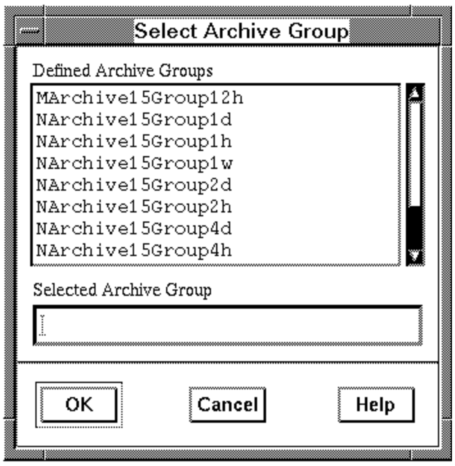
When you display this dialog box via the View Report Logs window, the Log Type field is not provided since the only valid log type is REPORTS. This is also the case when you use the View Production Data Logs window since the only valid type is PDL.



Log Class: Run Time
Log Type: Numeric Log

Figure 4-30. Retrieve Log Information by Location

When you choose **Retrieve by Group**, the Select Archive Group dialog box is displayed, Figure 4-31. This dialog box provides the means to display logs that belong to a specific archive group.



MArchive15Group12h
NArchive15Group1d
NArchive15Group1h
NArchive15Group1w
NArchive15Group2d
NArchive15Group2h
NArchive15Group4d
NArchive15Group4h

Figure 4-31. Retrieve by Group Dialog Box

b. Specify the appropriate information in the dialog box.

– To retrieve by location:

Specify the node in the Location field. Enter the History Manager name directly, or click on the **Modify** button to display a list of available History Managers.

Select the Log Class. The default class is Runtime. Runtime logs are the logs currently being stored in the History database. Select Runtime to archive runtime logs to the archive media.

Select the Log Type (Numeric, Message, or SPC). This step is not applicable for View Report Logs or View Production Data Logs.

– To retrieve by archive group, specify the Archive Group. Enter the archive group name directly, or click on the group name in the list of available archive groups.

c. Click on **OK** when you are finished with the dialog box.

This removes the dialog box and displays the appropriate logs in the View Log window. The read-only fields are filled in accordingly.

3. Select the logs that you want to archive from the list in the View Logs window. For Report logs, you select the entries (Reports) that you want to archive. For PDLs, you select the tasks that you want to archive.

To select one log at a time, click on the corresponding line in the View Logs window.

To select multiple contiguous logs, click on the first log you want to select, hold down <**Shift**>, and then click on the last log you want to select.

To select multiple logs that are not contiguous, hold down <**Ctrl**> and click on the logs you want to select.

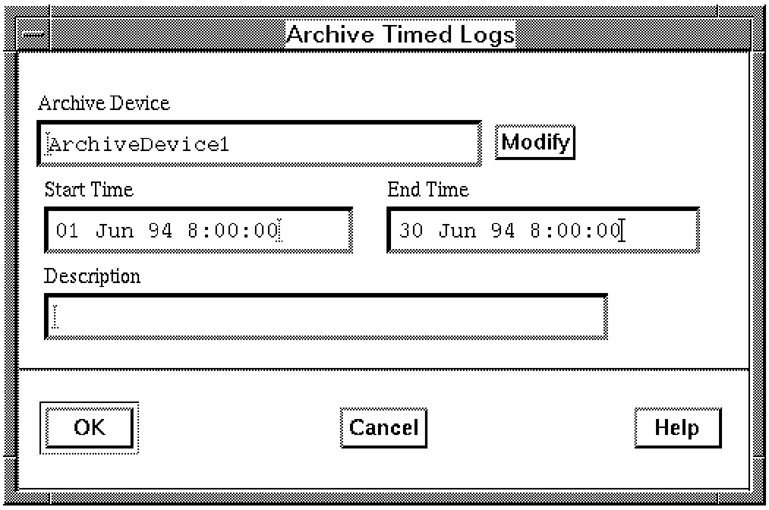
To de-select all selected logs, choose **Unselect All** from the **Select** menu.

NOTE

To facilitate the log selection process you can use the **Filter** and **Search** menu items from the **View** menu to search for a specific log and/or filter (reduce the number of) logs displayed based on log name, archive group, and/or log status. Refer to Section 4.5.4.2, View Archive Media Window - View Menu for details.

4. After you have selected the logs (or Reports, or tasks) to archive, choose **Archive Logs** from the **File** menu.

   This displays the Archive Timed Logs dialog box, Figure 4-32.



Archive Device
ArchiveDevice1

Start Time
01 Jun 94 8:00:00

End Time
30 Jun 94 8:00:00

Description

Figure 4-32. Archive Timed Logs

This dialog box provides the following fields for specifying archive parameters:

Archive Device	Enter the name of the archive device where you want to copy the logs to. Click on the **Modify** button by this field to display a list of available devices.
Start Time	Enter the date and time when you want the system to begin archiving log entries.
End Time	Enter the date and time when you want the system to stop archiving log entries.
Description	Enter an optional description to identify this archive entry.

5. Click on **OK** when you are finished with this dialog.

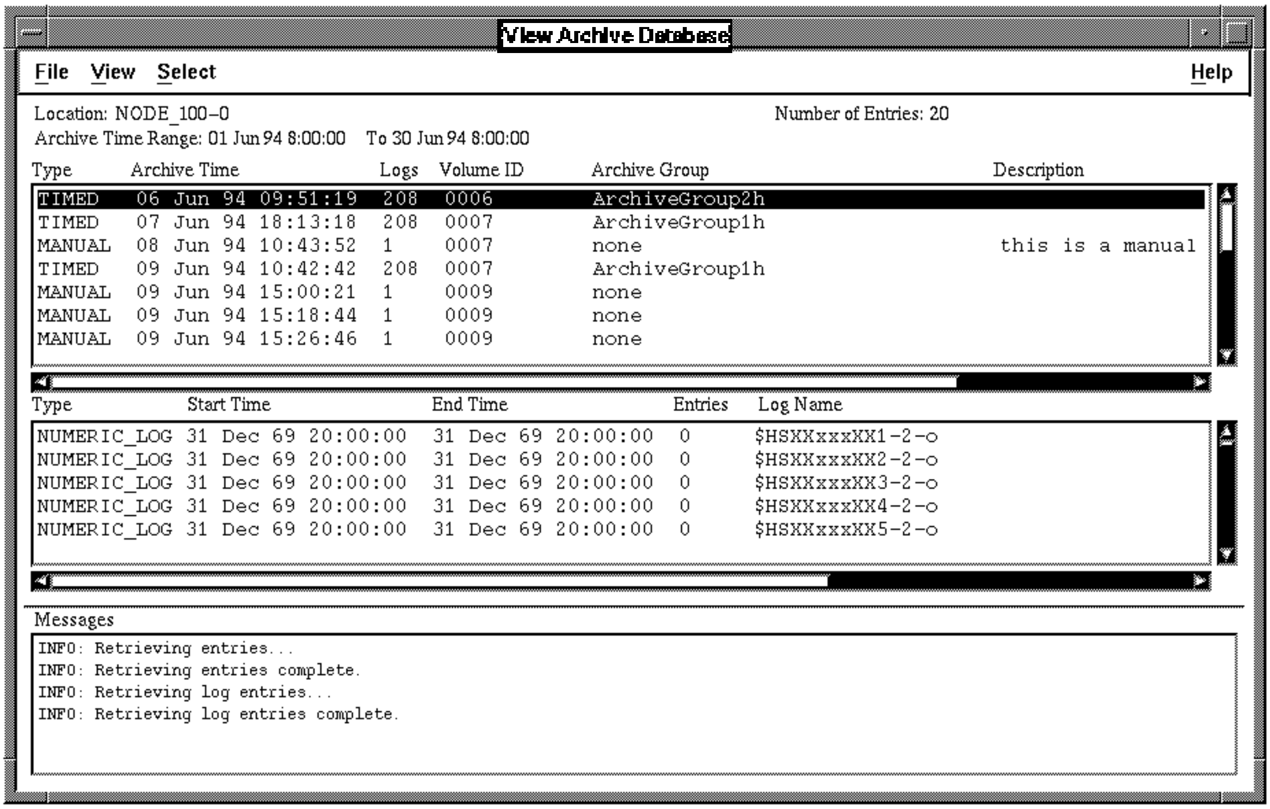
   The system makes an archive request to the specified archive device for data between the specified start and end times.

4.4.3.3 How to View Archive Database Information

When make an archive entry on the archive media, the archive entry is recorded in the archive database. The recorded information includes the archive time, volume ID, archive group, and so on.

The default tablespace for archive entries (HS_Archive) is 5 megabytes. This will hold approximately 30,000 entries. When the tablespace fills up, the oldest entries are deleted until enough room is available to insert the newest entries. If the maximum number of entries for the default size is less than what you will need, expand the tablespace as described in Section 5.6.7, How to Extend Tablespace for History Files.

You can retrieve this information and view it in the Archive Database window, Figure 4-33.



Location: NODE_100-0
Number of Entries: 20
Archive Time Range: 01 Jun 94 8:00:00 To 30 Jun 94 8:00:00

```
Type    Archive Time           Logs  Volume ID   Archive Group                    Description
TIMED   06 Jun 94 09:51:19     208   0006        ArchiveGroup2h
TIMED   07 Jun 94 18:13:18     208   0007        ArchiveGroup1h
MANUAL  08 Jun 94 10:43:52     1     0007        none                             this is a manual
TIMED   09 Jun 94 10:42:42     208   0007        ArchiveGroup1h
MANUAL  09 Jun 94 15:00:21     1     0009        none
MANUAL  09 Jun 94 15:18:44     1     0009        none
MANUAL  09 Jun 94 15:26:46     1     0009        none
```

```
Type         Start Time             End Time               Entries  Log Name
NUMERIC_LOG  31 Dec 69 20:00:00     31 Dec 69 20:00:00     0        $HSXXxxxXX1-2-o
NUMERIC_LOG  31 Dec 69 20:00:00     31 Dec 69 20:00:00     0        $HSXXxxxXX2-2-o
NUMERIC_LOG  31 Dec 69 20:00:00     31 Dec 69 20:00:00     0        $HSXXxxxXX3-2-o
NUMERIC_LOG  31 Dec 69 20:00:00     31 Dec 69 20:00:00     0        $HSXXxxxXX4-2-o
NUMERIC_LOG  31 Dec 69 20:00:00     31 Dec 69 20:00:00     0        $HSXXxxxXX5-2-o
```

```
INFO: Retrieving entries...
INFO: Retrieving entries complete.
INFO: Retrieving log entries...
INFO: Retrieving log entries complete.
```

Figure 4-33. View Archive Database Window

This information is useful when you want to restore archived logs from an archive media, and you do not know specifically which volume the logs are stored on. You can view the archive database to determine where logs are archived, mount the applicable volume on the archive device, and then restore the logs to the restored archive database.

This window is divided into four sections. The top section has the following read-only fields:

Location	Node where the archive database resides.
Archive Time Range	Start and end time for the entries you want to retrieve
Number of entries	Number of entries retrieved.

The information in these fields is determined by the retrieved archive database info dialog described later in this section.

The upper middle section displays the following information for each archive entry:

Type	Type of archive entry: TIMED or MANUAL as defined in Basis field of Archive Group.
Archive Time	Time when the archive entry was recorded on the archive device.
Logs	Number of logs in the archive entry.
Volume ID	The volume ID for the tape the archive entry is recorded on.
Archive Group	Indicates the Archive Group that the logs in this archive entry belong to if the entry was made by archive group. If the entry was made by location, the entry in this field is None.
Description	Archive entry description as defined in the Archive Timed Logs dialog box.

The lower middle section displays the logs in a selected archive entry. The information provided for each log is:

Type	Log type: NUMERIC, MESSAGE,SPC, PDL, or REPORT.
Start Time	Start time for archive data for this log.
End Time	end time for archive data for this log.
Entries	Number of samples archived for this log.
Log Name	Log name.

The bottom section displays messages for the activities you perform in this window.

To open the Archive Database window and view archive database entries:

1. In the AdvaInform History window, choose **Archive** from the **Tools** menu, and then choose **View Archive Database** from the cascade menu.

   This opens the View Archive Database window. Initially, the fields are blank (un-defined).

2. Specify a list or range of archive entries.

   a. Choose **Retrieve Archive Database Info** from the **File** menu.

This displays the Retrieve Archive Database Entries dialog box, Figure 4-34.

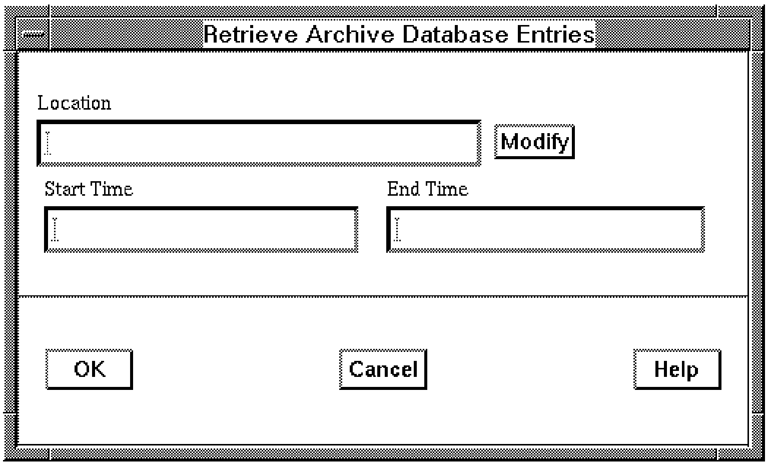


Figure 4-34. Retrieve Archive Database Entries Dialog Box

This dialog box provides the means to display a list of archive database entries for a specific location and time range.

b. Specify the node in the Location field. Click on the **Modify** button to display a list of available History Managers.

c. Specify the time range (start and end times) to retrieve data for.

d. Click on **OK**.

This displays archive database entries for the specified location and time range.

3. Select a database entry to display the corresponding logs.

NOTE

To facilitate the database entry selection process, you can use menu items in the **View** menu to search for a specific entry, or to filter (reduce the number of) displayed entries. To do so, use the appropriate **View** menu items as described in Section 4.5.4.2, View Archive Media Window - View Menu.

When you select an entry, the corresponding logs are displayed in the lower middle portion of the Archive Database window.

4.4.3.4 How to Restore Archive Logs from Archive Media to Restored History Database

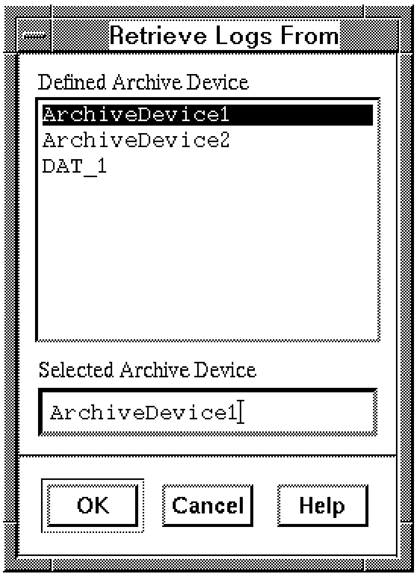
To restore archive logs from the archive media to the restored history database:

1. In the AdvaInform History window, choose **Archive** from the **Tools** menu, and then choose **View Archive Media** from the cascade menu.

   This opens the View Archive Media window.

2. Choose **Retrieve Archive Entries** from the **File** menu.

   This displays the Retrieve Logs From dialog box, Figure 4-35. This dialog box provides a list of archive devices to choose from.



Defined Archive Device

ArchiveDevice1
ArchiveDevice2
DAT_1

Selected Archive Device: ArchiveDevice1

Figure 4-35. Retrieve Logs From Dialog Box

3. Select an archive device from the list, and then click on **OK**.

   This displays archive entries stored on the tape in the specified archive device.

4. Select the entry that you want to restore to the database.

   NOTE

   To facilitate the archive entry selection process, you can use menu items in the **View** menu to search for a specific entry, or to filter (reduce the number of) displayed entries. To do so, use the appropriate **View** menu items as described in Section 4.5.4.2, View Archive Media Window - View Menu.

   When you select an archive entry, the corresponding logs are displayed in the lower middle portion of the View Archive Media window.

5. Select the logs that you want to restore.

   You can select one log at a time by clicking on the corresponding line.

   To select multiple contiguous logs, click on the first log you want to select, hold down <**Shift**>, and then click on the last log you want to select.

   To select multiple logs that are not contiguous, hold down <**Ctrl**> and click on the logs you want to select.

   To de-select all selected logs, choose **Unselect All** from the **Select** menu.

   To facilitate the log selection process, you can use menu items in the **View** menu to search for a specific log, or to filter (reduce the number of) displayed logs. To do so, use the appropriate **View** menu items as described in Section 4.5.4.2, View Archive Media Window - View Menu.

6. To restore the selected logs, choose **Restore from Archive Database** from the **File** menu.

NOTE

Only log entries that were archived at the same time can be restored in a given restore operation. Two or more restore operations are required to restore log entries that were archived at different times.

4.4.3.5 How to Delete Archive Entries from the Archive Database

To delete archive entries from the archive database, follow the instructions in Section 4.4.3.3, How to View Archive Database Information, and then choose **Delete from Archive Database** from the **File** menu in the View Archive Database window.

4.4.3.6 How to Delete Restored Logs

The View Logs window, Figure 4-27, provides the means to delete logs from the restored portion of the database.To delete restored logs:

1. In the AdvaInform History window, choose **Archive** from the **Tools** menu, and then choose **View Logs** from the cascade menu.

   This opens the View Logs window. Initially, the fields are blank (un-defined).

2. Specify a list or range of logs from which you can select the logs you want to archive.

   a. Choose **Retrieve Logs** from the **File** menu.

      This displays a cascade menu where you can choose either **Retrieve by Location** or **Retrieve by Group**.

      The only way to retrieve restored logs is by location. Therefore, choose **Retrieve by Location**.

When you choose **Retrieve by Location**, the Retrieve Log Information by Location dialog box is displayed, Figure 4-36. This dialog box provides the means to display a list of logs for a specific node (Location), based on Log Class and Log Type.

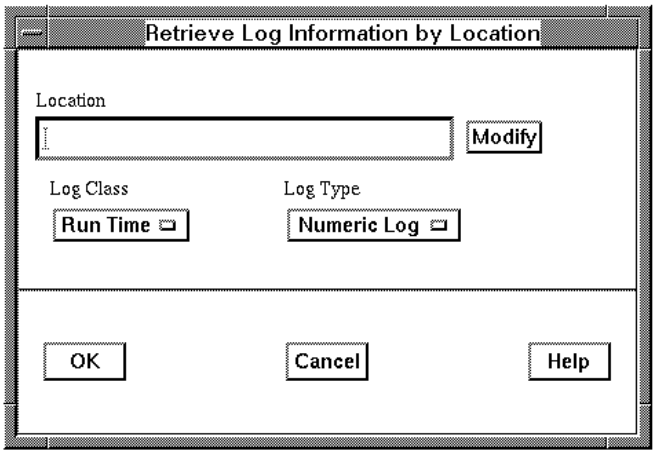


Figure 4-36. Retrieve Log Information By Location Dialog Box

b. Specify the following information in the dialog box.

In the Location field, enter the History Manager name directly, or click on the **Modify** button to display a list of available History Managers.

Select Restored for Log Class.

Select the Log Type (Numeric, or Message; SPC and PDL are also available if the corresponding options are installed on your system).

c. Click on **OK** when you are finished with the dialog box.

3. Select the logs that you want to delete from the list in the Archive Logs window.

You can select one log at a time by clicking on the corresponding line.

To select multiple contiguous logs, click on the first log you want to select, hold down <**Shift**>, and then click on the last log you want to select.

To select multiple non-contiguous logs, hold down <**Ctrl**> and click on the logs you want to select. To de-select all selected logs, choose **Unselect All** from the **Select** menu.

NOTE

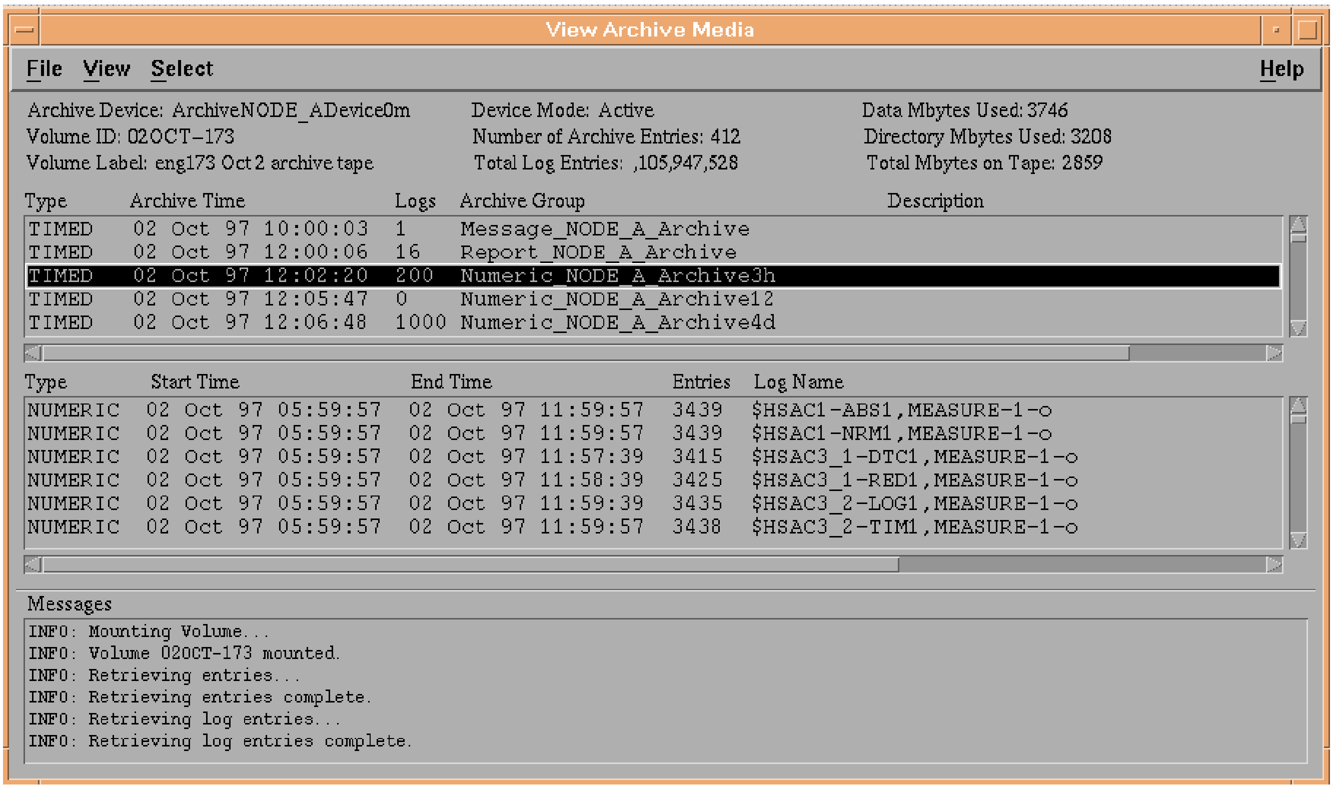
To facilitate the log selection process you can use the **Filter** and **Search** menu items from the **View** menu to search for a specific log and/or filter (reduce the number of) logs displayed based on log name, archive group, and/or log status. Refer to Section 4.5.4.2, View Archive Media Window - View Menu for details.

4. Choose **Delete Restored Logs** from the **File** menu. This displays a message box to confirm or cancel the delete command.

5. Click on **Delete** to continue the delete function or **Cancel** to cancel it.

4.4.3.7 How to Add Database Entries from Archive Media to History Database

When you restore logs from an Archive Media to the History database, it is helpful to have the corresponding database information for those logs (Volume ID, Label, start and stop times and so on). The View Archive Media window, Figure 4-37, provides the means to copy database entries from the Archive Media to the archive database.



Archive Device: ArchiveNODE_ADevice0m
Volume ID: 02OCT–173
Volume Label: eng173 Oct 2 archive tape
Device Mode: Active
Number of Archive Entries: 412
Total Log Entries: ,105,947,528
Data Mbytes Used: 3746
Directory Mbytes Used: 3208
Total Mbytes on Tape: 2859

Type	Archive Time	Logs	Archive Group	Description
TIMED	02 Oct 97 10:00:03	1	Message_NODE_A_Archive	
TIMED	02 Oct 97 12:00:06	16	Report_NODE_A_Archive	
TIMED	02 Oct 97 12:02:20	200	Numeric_NODE_A_Archive3h	
TIMED	02 Oct 97 12:05:47	0	Numeric_NODE_A_Archive12	
TIMED	02 Oct 97 12:06:48	1000	Numeric_NODE_A_Archive4d	

Type	Start Time	End Time	Entries	Log Name
NUMERIC	02 Oct 97 05:59:57	02 Oct 97 11:59:57	3439	$HSAC1-ABS1,MEASURE-1-o
NUMERIC	02 Oct 97 05:59:57	02 Oct 97 11:59:57	3439	$HSAC1-NRM1,MEASURE-1-o
NUMERIC	02 Oct 97 05:59:57	02 Oct 97 11:57:39	3415	$HSAC3_1-DTC1,MEASURE-1-o
NUMERIC	02 Oct 97 05:59:57	02 Oct 97 11:58:39	3425	$HSAC3_1-RED1,MEASURE-1-o
NUMERIC	02 Oct 97 05:59:57	02 Oct 97 11:59:39	3435	$HSAC3_2-LOG1,MEASURE-1-o
NUMERIC	02 Oct 97 05:59:57	02 Oct 97 11:59:57	3438	$HSAC3_2-TIM1,MEASURE-1-o

Messages

```
INFO: Mounting Volume...
INFO: Volume 02OCT-173 mounted.
INFO: Retrieving entries...
INFO: Retrieving entries complete.
INFO: Retrieving log entries...
INFO: Retrieving log entries complete.
```

Figure 4-37. Archive Media Window

This window is divided into four sections. The top section has five read-only fields whose information is determined by the Retrieve Log information dialog (described later in this section):

Archive Device	This is the name of the archive device where the archive tape (media) is installed.
Volume ID	This is a name of this specific tape.
Volume Label	This is an optional descriptor to further identify the tape.
Device Mode	Active or Inactive.
Number of Archive Entries	This is the number of occasions when data have been archived to this tape.

Total Log Entries	This is the total number of archived log entries.
Data Mbytes Used	This is the amount of log data in megabytes stored on tape.
Directory Mbytes Used	This is the amount of directory/catalog data in megabytes stored on tape.
Total Mbytes on Tape	This is the total amount of data (log data and directory data combined) in megabytes stored on tape. If the total for Data and Directory combined exceeds 4 gigabytes, this number is the quantity in excess of 4 gigabytes. For example, in Figure 4-37, (3746 + 3208) - 4096 = 2858. By looking at the number of full tapes, the Total Mbytes value can be used to estimate how much data will fit on a tape, and approximate when the tape will be full.

The upper middle section displays a list archive entries according to the Retrieve Log Information dialog. An archive entry represents all the logs archived as a result of a specific archive request. The information provided for each archive entry is:

Type	How archive was initiated (TIMED or MANUAL).
Archive Time	When the archive entry occurred.
Logs	Number of logs that were archived.
Archive Group	Archive Group that the logs belong to.
Description	Description as specified in the Archive Timed Logs dialog box.

The lower middle section displays the logs in a selected archive entry. The information provided for each log is:

Type	Type of log.
Start Time	Start time for archived data.
End Time	End time for archived data.
Entries	Number of entries archived for this log.
Log Name	Log name.

The bottom section displays messages for the activities you perform in this window.

To add database entries to the archive database:

1. In the AdvaInform History window, choose **Archive** from the **Tools** menu, and then choose **View Archive Media** from the cascade menu.

   This opens the Archive Media window.

2. Choose **Retrieve Archive Entries** from the **File** menu.

This displays the Retrieve Logs From dialog box. This dialog box provides a list of archive devices to choose from.

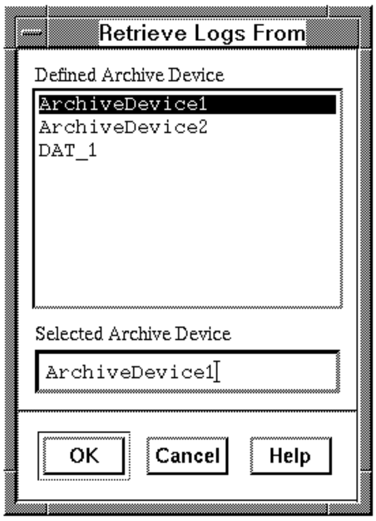


Figure 4-38. Retrieve Logs From I Dialog Box

3. Select an archive device from the list, and then click on **OK**.

   This displays archive entries stored on the tape in the specified archive device.

4. Select the entry that you want to add to the database.

NOTE

To facilitate the archive entry selection process, you can use menu items in the **View** menu to search for a specific entry, or to filter (reduce the number of) displayed entries. To do so, use the appropriate **View** menu items as described in Section 4.5.4.2, View Archive Media Window - View Menu.

When you select an archive entry, the corresponding logs are displayed in the lower middle portion of the Archive Media window.

5. To add the entry to the database, choose **Add Entries to Database** from the **File** menu.

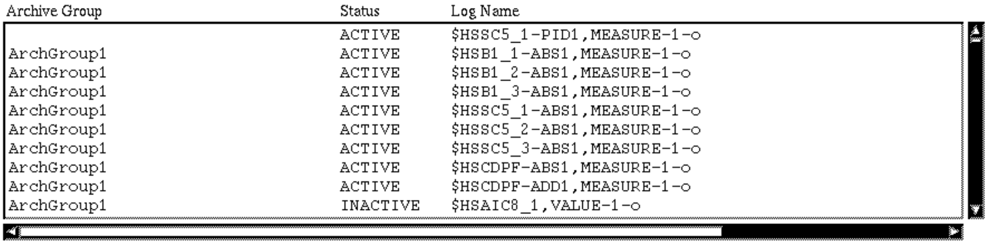
4.4.3.8 How to Use the Filter and Search Dialogs

When you retrieve archive entry and/or log information in either the View Log, View Archive Media, or View Archive Database windows, you can use the **View** menu items to facilitate the process of selecting an archive entry or logs. You can use Filter to reduce the displayed list of entries (or logs), or you can use Search to scroll directly to a specific entry (or log).

Operation of the Filter and Search menu items is very similar for each of the three windows. To demonstrate, this section provides an example of the **Filter** menu item in the View Log window. For details regarding operation of a specific Filter or Search menu item, refer to the applicable subsection in Section 4.5, Runtime Menus.

Example, Apply Filter to List of Logs in Archive Log Window

This example shows how to reduce a long list of Logs in the Archive Log window, Figure 4-39.



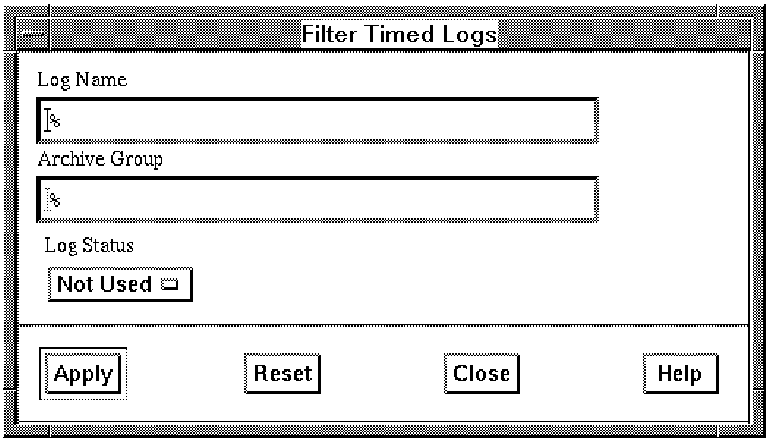
```
Archive Group                   Status      Log Name
                                ACTIVE      $HSSC5_1-PID1,MEASURE-1-o
ArchGroup1                      ACTIVE      $HSB1_1-ABS1,MEASURE-1-o
ArchGroup1                      ACTIVE      $HSB1_2-ABS1,MEASURE-1-o
ArchGroup1                      ACTIVE      $HSB1_3-ABS1,MEASURE-1-o
ArchGroup1                      ACTIVE      $HSSC5_1-ABS1,MEASURE-1-o
ArchGroup1                      ACTIVE      $HSSC5_2-ABS1,MEASURE-1-o
ArchGroup1                      ACTIVE      $HSSC5_3-ABS1,MEASURE-1-o
ArchGroup1                      ACTIVE      $HSCDPF-ABS1,MEASURE-1-o
ArchGroup1                      ACTIVE      $HSCDPF-ADD1,MEASURE-1-o
ArchGroup1                      INACTIVE    $HSAIC8_1,VALUE-1-o
```

Figure 4-39. Lengthy Listing of Logs in Archive Log Window

Filter reduces the number of logs displayed in the Archive Log window based on a user-defined combination of log name, archive group, and log status. To apply a filter to this list:

1. Choose **Filter** from the **View** menu.

   This displays the Filter Timed Logs dialog box, Figure 4-57.



Log Name: %

Archive Group: %

Log Status: Not Used

Figure 4-40. Filter Logs Dialog

2. Enter the filter parameters as required.

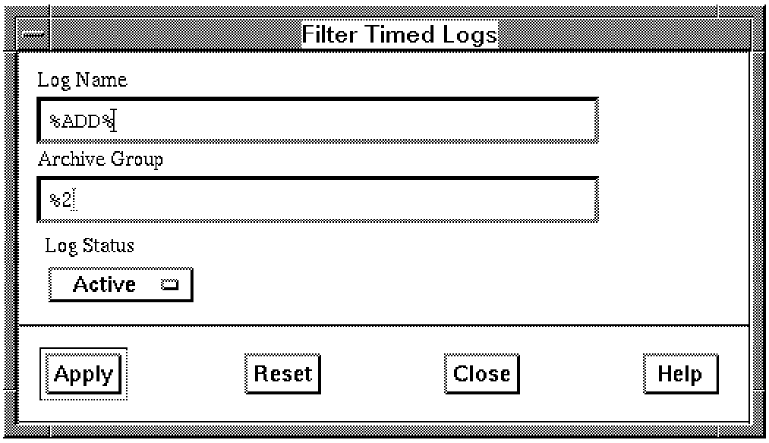
   You can use the parameters described below in any combination.

   Log Name - Only display logs whose name has the specified text string.

   Archive Group - Only display logs that belong to the specified archive group.

   Log Status - Only display logs that have the specified log status.

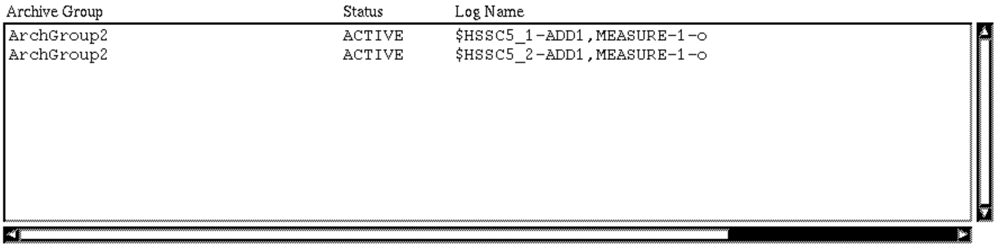
The % character is a wildcard. It can used as a substitute for any character string, and can be used in the beginning, end, and/or middle of a character string. An example is shown in Figure 4-41.



Log Name: %ADD%
Archive Group: %2
Log Status: Active

Figure 4-41. Example of Using % (wildcard) Character

This example reduces the displayed list of logs to only logs that have the character string "ADD", that belong to an archive group whose name ends with the character "2", and whose log status is ACTIVE, Figure 4-42. Note that all three criteria must be met.



```
Archive Group                          Status       Log Name
ArchGroup2                             ACTIVE       $HSSC5_1-ADD1,MEASURE-1-o
ArchGroup2                             ACTIVE       $HSSC5_2-ADD1,MEASURE-1-o
```

Figure 4-42. Result of Filter

4.4.4 Handling Production Data Logs (MOD 300)

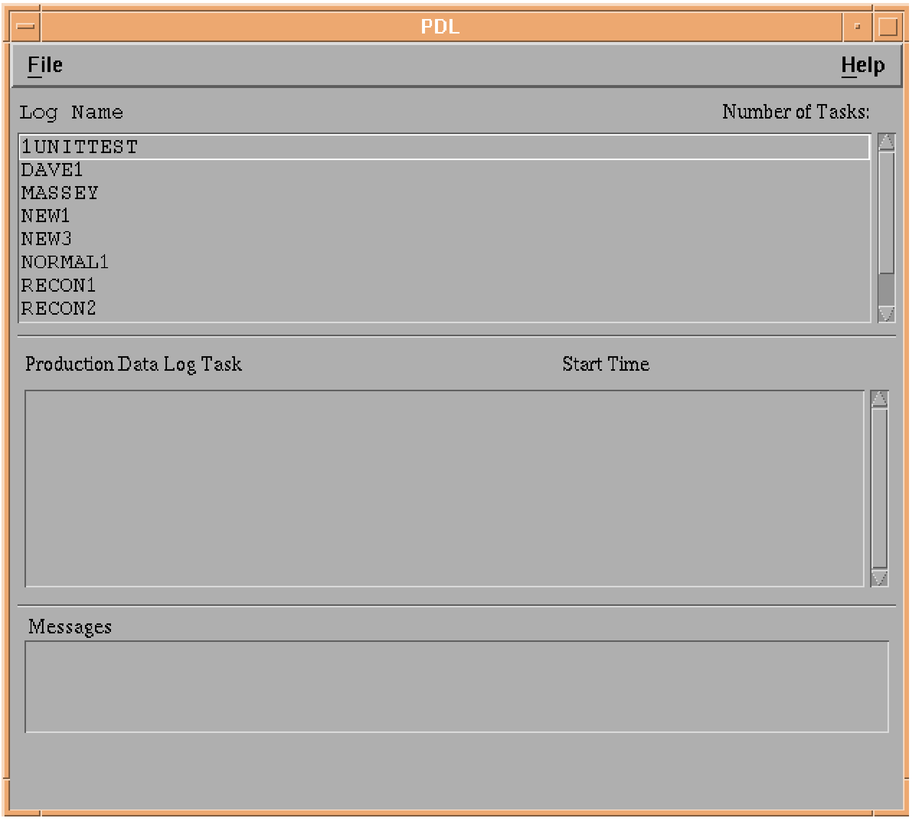
This section describes how to view and delete Production Data Logs (PDLs), and delete tasks at the job and batch levels.

A PDL is created in History whenever a START JOB message is received from Batch 300, or whenever a START Batch message is received and the batch is not part of a job. Therefore the root task in the PDL hierarchy can either be a job or a batch (refer to *Batch 300 User's Guide* for a description of PDL structure).

4.4.4.1 How to Display Jobs and Batches

To display jobs and batches:

1. Choose **PDL Logs - Delete PDLs** from the **Tools** menu in the History Configuration window. This opens the PDL window, Figure 4-43.

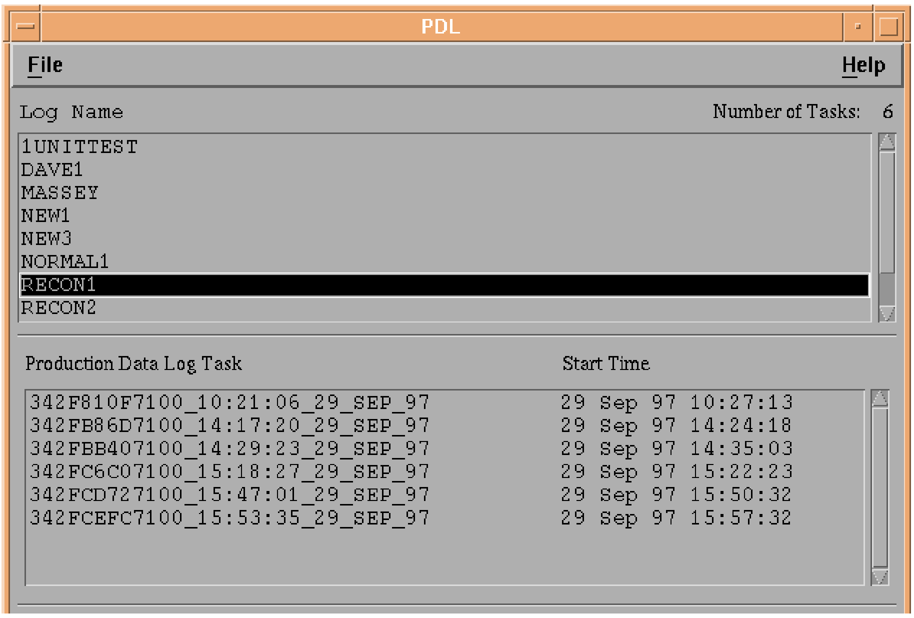


1UNITTEST
DAVE1
MASSEY
NEW1
NEW3
NORMAL1
RECON1
RECON2

Figure 4-43. PDL Window

PDLs are listed in the Log Name list. These PDLs represent level 0 jobs (when PDL is created by START JOB message from Batch 300) or level 1 batches (when PDL is created by START BATCH message from TCL and the batch is not part of a job). The PDL name is derived from the name of the root task.

2. Select a PDL from the list. This displays the related tasks for the selected PDL in the Production Data Log Task field, Figure 4-44.



Log Name

Number of Tasks: 6

1UNITTEST
DAVE1
MASSEY
NEW1
NEW3
NORMAL1
RECON1
RECON2

Production Data Log Task	Start Time
342F810F7100_10:21:06_29_SEP_97	29 Sep 97 10:27:13
342FB86D7100_14:17:20_29_SEP_97	29 Sep 97 14:24:18
342FBB407100_14:29:23_29_SEP_97	29 Sep 97 14:35:03
342FC6C07100_15:18:27_29_SEP_97	29 Sep 97 15:22:23
342FCD727100_15:47:01_29_SEP_97	29 Sep 97 15:50:32
342FCEFC7100_15:53:35_29_SEP_97	29 Sep 97 15:57:32

Figure 4-44. PDL Window with Tasks Displayed

If the selected PDL is a batch task, that same task is displayed in the Production Data Log Task field, and the number of Tasks indicated will be 1.

4.4.4.2 How to Delete a PDL and/or Batch Task

You can delete PDLs and batch tasks via the PDL window. When you delete a PDL or batch task, all related tasks are also deleted (batch-, unit procedure-, and phase-level).

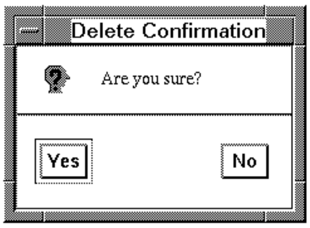
NOTE

When you delete a PDL, all batch, unit procedure, and phase tasks under that PDL are deleted, and the object is deleted from History. When you delete a batch task, all unit procedure, and phase tasks under that batch are deleted Also, all log and resource associations belonging to the job or batch are deleted.

To delete a PDL from the PDL window:

1. Select the PDL or PDLs from the Log Name field. To select a single PDL, simply click once on the corresponding line. To select multiple non-contiguous PDLs, hold down the **ctrl** key as you click. To select multiple contiguous PDLs, hold down the **Shift** key, click on the first PDL to be deleted, then click on the last PDL to be deleted. To deselect a selected PDL, click on it again.
2. Choose **Delete PDLs** from the **File** menu.

This displays a confirmation message, Figure 4-45.



Are you sure?

Figure 4-45. Delete Confirmation

3. Click on **Yes** to confirm or **No** to cancel.

To delete a batch task from the PDL window:

1. Select the PDL whose batch tasks you want to delete. In this case, you can only select one PDL.
2. Select one or more batch tasks to delete. To select a single task, simply click once on the corresponding line. To select multiple non-contiguous tasks, hold down the **ctrl** key as you click. To select multiple contiguous tasks, hold down the **Shift** key, click on the first task to be deleted, then click on the last task to be deleted. To deselect a selected task, click on it again.
3. Choose **Delete Tasks** from the **File** menu.

   This displays a confirmation message, Figure 4-45.
4. Click on **Yes** to confirm or **No** to cancel.

4.5 Runtime Menus

4.5.1 Action Menu in Log Set and Log Group Windows

The **Action** menu of the Log Set and Log Group windows, Figure 4-46, is used to activate and de-activate all logs in the group or set. Active logs collect data, while inactive logs do not.

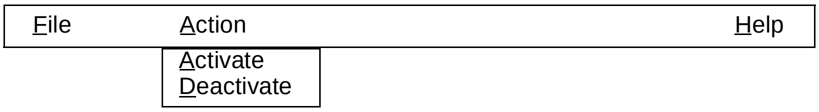


Figure 4-46. Action Menu of the Log Set and Log Group Windows

4.5.2 Menus for Initialize Archive Media Window

The Initialize Archive Media window is opened when you choose **Initialize Archive Media** from the **Tools** menu on the History Builder menu bar. This window is used to initialize a tape loaded in the archive device. The menu bar for this window is shown in Figure 4-48.

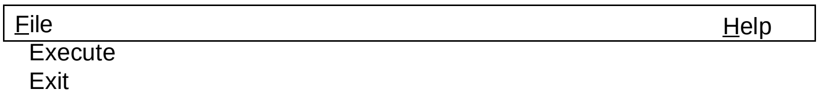


Figure 4-47. Initialize Media Menu Bar

File - Execute

Initialize the media according to the entries you made in this window.

File - Exit

Close the Initialize Archive Media window.

4.5.3 Menus for the View Logs Window

The View Logs window is opened when you choose **View Logs** from the **Tools**-**Archive** menu on the History Builder menu bar. This window provides access to a directory of historical logs that reside on the long term storage media. The window is empty until you specify what data to retrieve via the **Retrieve Log Info** menu item in the **File** menu. The menu bar for the Archive Log window is shown in Figure 4-48.

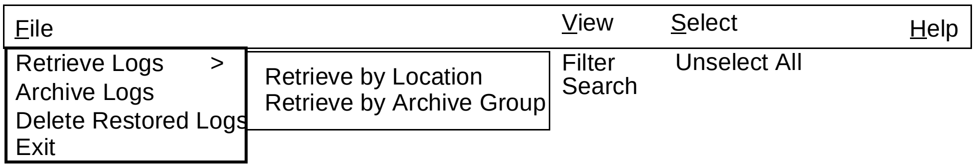


Figure 4-48. Archive Logs Menu Bar

4.5.3.1 View Logs Window - File Menu

File - Retrieve Logs

Retrieve Logs is used to specify the information to be displayed in the View Log window. This menu item provides a cascade with two options:

- **Retrieve by Location**
- **Retrieve by Archive Group**

When you select one of these options, the corresponding dialog box is displayed. The operation of these dialog boxes is described in Section 4.4.3.2, How to Archive Logs. When you retrieve by location, you must choose a log class (runtime or restored) to retrieve. Runtime logs are logs that are currently stored in the History database. You retrieve runtime logs when you want to archive logs in the History database. Restored logs are logs that were archived and have been restored from the archive device to the restored database. You retrieve restored logs when you want to delete one or more logs from the restored database.

File - Archive Logs

Archive Logs is used to archive logs that you have retrieved from the runtime History database and selected in the View Logs window. This menu item is only available when you select the Runtime log class in the respective Retrieve Logs by (Location or Archive Group) dialog box. When you choose **Archive Logs**, the Archive Timed Logs dialog box is displayed. Operation of this dialog box is described in Section 4.4.3.2, How to Archive Logs.

File - Delete Restored Logs

Delete Restored Logs is used to delete logs that you have retrieved from the restored History database and selected in the View Logs window. This menu item is only available when you select the Restored log class in the respective Retrieve Logs by (Location or Archive Group) dialog box. When you choose **Delete Restored Logs**, a message box is displayed to allow you to confirm or cancel the delete function.

File - Exit

Exit is used to close the View Logs window.

4.5.3.2 View Logs Window - View Menu

The **View** menu for the Archive Log window is shown in Figure 4-49.

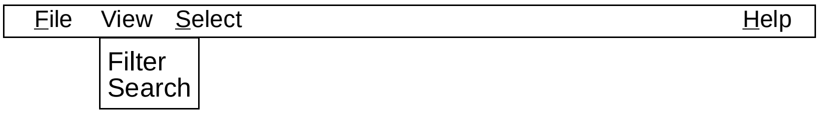
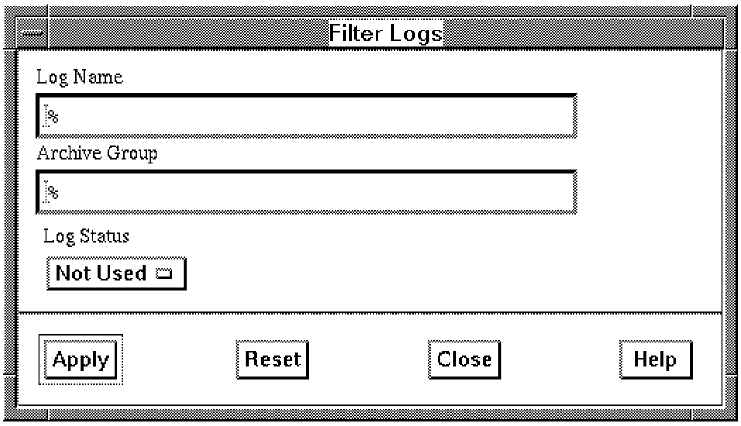


Figure 4-49. View Menu Items

View - Filter

Filter is used to reduce the number of logs displayed in the window by applying a filter based on a user-defined combination of log name, archive group, and log status. When you choose **Filter**, the Filter Logs dialog box is displayed, Figure 4-57.



Log Name: %

Archive Group: %

Log Status: Not Used

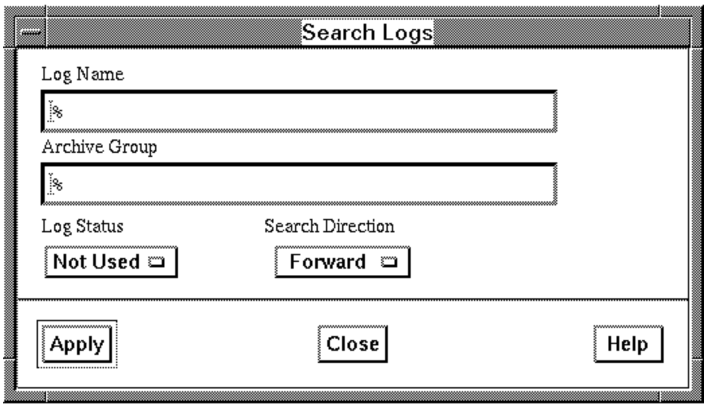
Figure 4-50. Filter Logs Dialog

The filter parameters are described below. You can use these parameters in any combination as required.

Log Name	Only display logs whose name has the specified text string. % is a wild card text string which indicates any log.
Archive Group	Only display logs that belong to the specified archive group. % is a wild card text string which indicates any archive group.
Log Status	Only display logs that have the specified log status. Not Used means that both active and inactive logs are displayed.

View - Search

Search is used to search for a specific log. It scrolls the list so that the first log in the list that matches the specified parameters is at the top. When you choose **Search**, the Search Logs dialog box is displayed, Figure 4-58. The parameters are described below.



Log Name
%
Archive Group
%
Log Status: Not Used
Search Direction: Forward

Figure 4-51. Search Logs Dialog

Log Name	Search for log whose name has the specified text string. % is a wild card text string which indicates any log.
Archive Group	Search for log that belongs to the specified archive group. % is a wild card text string which indicates any archive group.
Log Status	Search for log that has the specified log status.
Search Direction	Search in the forward or backward direction.

4.5.3.3 View Logs Window - Select Menu

The **Select** menu for the View Logs window is shown in Figure 4-61.

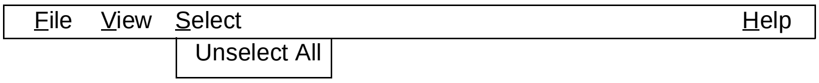


Figure 4-52. Select Menu

Select - Unselect All

This menu item unselects all selected logs in the View Logs window.

4.5.4 Menus for View Archive Media Window

The View Archive Media window is opened when you choose **View Archive Media** from the **Tools** menu on the History Builder menu bar.

4.5.4.1 View Archive Media Window - File Menu

The **File** menu for the Archive Media window is shown in Figure 4-53 (MOD 300) and Figure 4-54 (Master).

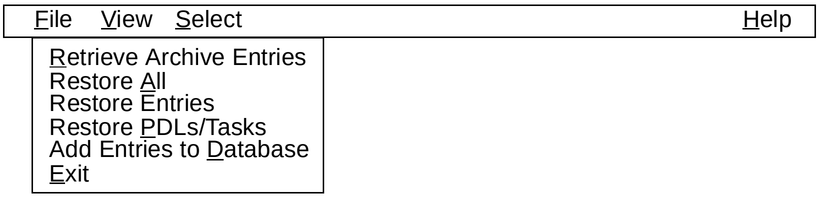


Figure 4-53. Archive Media - File Menu

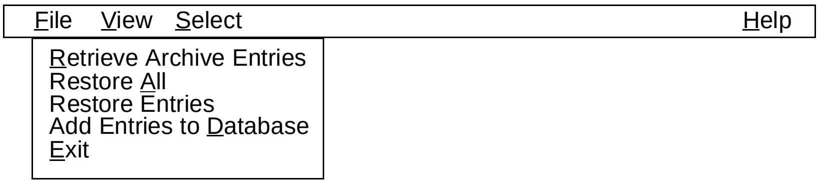
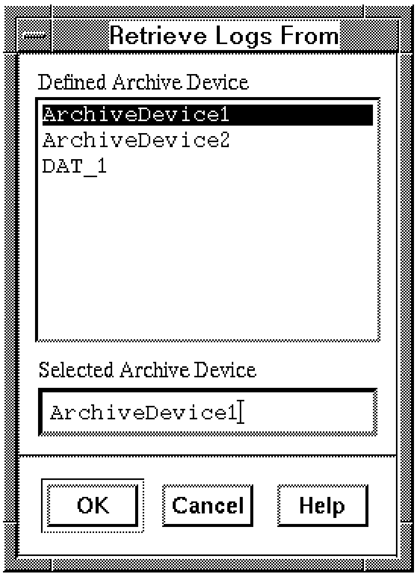


Figure 4-54. Archive Media - File Menu

File - Retrieve Archive Entries

Retrieve Archive Entries is used to retrieve archive entries from the storage media for viewing. This menu item displays the Retrieve Logs From dialog box, Figure 4-58. This dialog box is used to specify the archive device that you want to retrieve archive entries from.



ArchiveDevice1

Figure 4-55. Retrieve Logs From Dialog Box

File - Restore All

Restore All is used to restore all logs in the selected archive entry.

File - Restore Entries

Restore Entries is used to restore selected archived logs from the archive media to the History database.

File - Restore PDLs/Tasks

This menu item is only available when the PDL option is installed on your system. It is used to restore selected PDLs.

File - Add Entries to Database

When you restore logs from the archive media to the History database, it is helpful to have the corresponding database information for those logs (Volume ID, Label, start and stop times and so on). The procedure to add these entries to the History database is the same as the procedure to restore logs from the DAT drive (as described in Section 4.4.3.4, How to Restore Archive Logs from Archive Media to Restored History Database). The only difference is that you use the **Add Entries to Database** menu item rather than the **Restore Entries** menu item.

File - Exit

Exit is used to close the View Archive Media window.

4.5.4.2 View Archive Media Window - View Menu

The **View** menu for the View Archive Media window is shown in Figure 4-56.

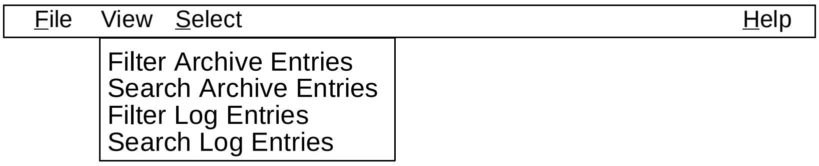
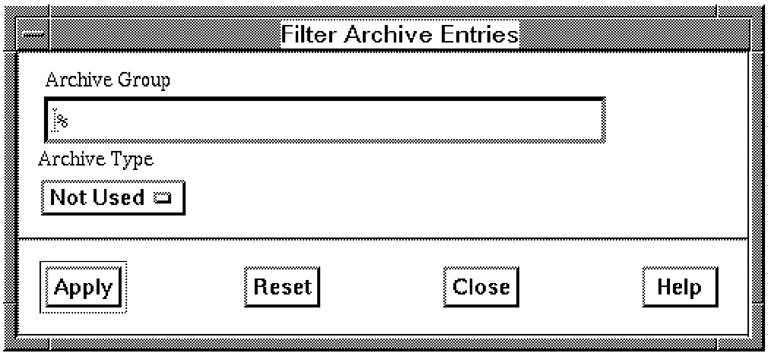


Figure 4-56. View Menu Items

View - Filter Archive Entries

Filter Archive Entries is used to reduce the number of archive entries displayed by applying a filter based on a user-defined combination of archive group and archive type. When you choose **Filter Archive Entries**, the Filter Archive Entries dialog box is displayed, Figure 4-57.



Archive Group
%

Archive Type
Not Used

Figure 4-57. Filter Entries Dialog

The filter parameters are described below. You can use these parameters in any combination as required.

Archive Group	Only display archive entries whose archive group is specified in this field. % is a wild card text string.
Archive Type	Only display archive entries whose archive type is as specified in this field (MANUAL or TIMED).

View - Search Archive Entries

Search Archive Entries is used to search for a specific archive entry. This function finds the first archive entry in the displayed list that matches the specified parameters, and scrolls the list so that archive entry is at the top. When you choose **Search Archive Entries**, the Search Archive Entries dialog box is displayed, Figure 4-58. The parameters are described below.

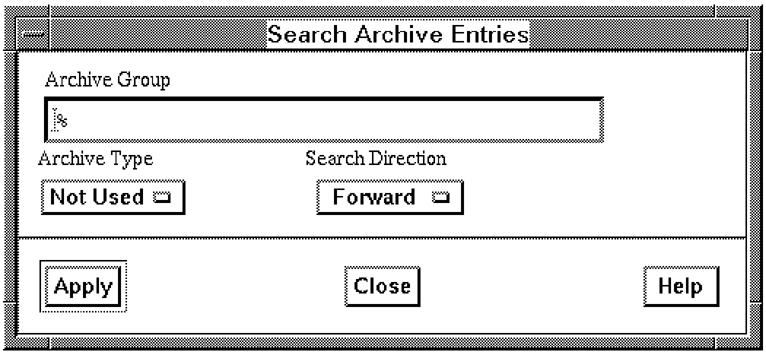
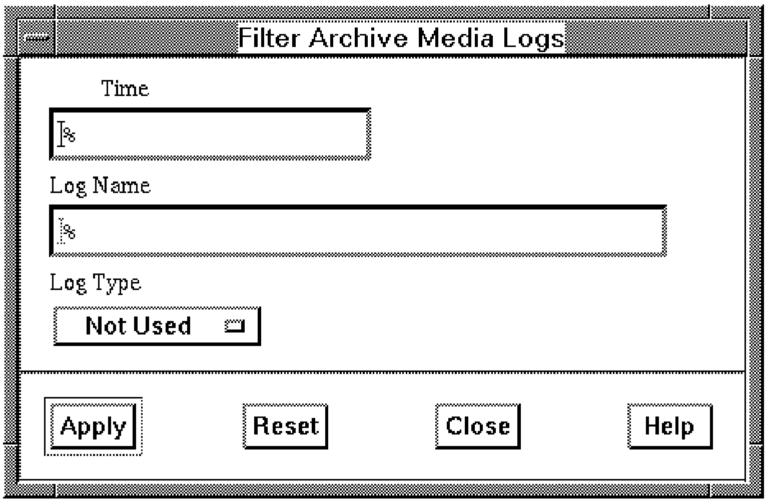


Figure 4-58. Search Logs Dialog

Archive Group	Search for archive entries that belong to specified archive group. % is a wild card text string.
Archive Type	Search for entries of specified archive type (Manual or Timed).
Search Direction	Displays the selected search direction (forward or backward). To change the search direction, click on the right hand button.

View - Filter Log Entries

Filter Log Entries is used to reduce the number of archive entries displayed by applying a filter based on a user-defined combination of log name, log type, and time range. When you choose **Filter Log Entries**, the Filter Archive Media Logs dialog box is displayed, Figure 4-57.



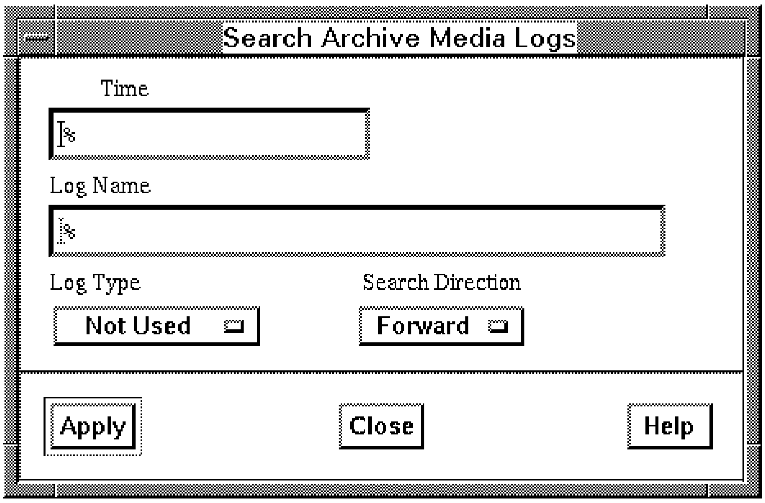
Time
%
Log Name
%
Log Type
Not Used

Figure 4-59. Filter Archive Media Log s Dialog

Log Name	Only display logs whose name has the specified text string. % is a wild card text string.
Log Type	Only display logs whose log type is as specified in this field.
Time	Only display logs whose archive start and end times cover the time specified in this field. For example, if the log start time is 30 aug 95 10:00:00 and the end time is 30 aug 95 12:00:00, and the filter time entered in this dialog is 30 aug 95 10:45:00, the log will be displayed. The format for time is as follows: day of the month, month (first three letters), year (last two digits), hours:minutes:seconds. For example: `07 Jun 95 09:35:00`.

View - Search Log Entries

Search Archive Entries is used to search for a specific archive entry. This function finds the first archive entry in the displayed list that matches the specified parameters, and scrolls the list so that archive entry is at the top. When you choose **Search Archive Entries**, the Search Archive Media Logs dialog box is displayed, Figure 4-58. The parameters are described below.



Time: %

Log Name: %

Log Type: Not Used

Search Direction: Forward

Figure 4-60. Search Archive Media Logs Dialog

Log Name	Search for logs whose name has the specified text string. % is a wild card text string.
Log Type	Search for logs whose log type is as specified in this field.
Time	Only display logs whose start and end times cover the time specified in this field. For example, if the log start time is 30 aug 95 10:00:00 and the end time is 30 aug 95 12:00:00, and the filter time entered in this dialog is 30 aug 95 10:45:00, the log will be displayed. The format for time is as follows: day of the month, month (first three letters), year (last two digits), hours:minutes:seconds. For example: `07 Jun 95 09:35:00`.
Search Direction	Displays the selected search direction (forward or backward). To change the search direction, click on the right hand button.

4.5.4.3 View Archive Media Window - Select Menu

The **Select** menu for the Archive Media window is shown in Figure 4-61.

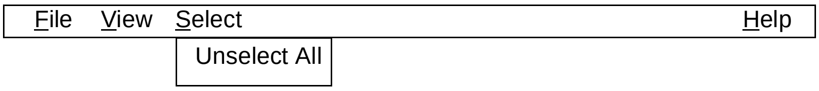


Figure 4-61. Select Menu

Select - Unselect All

This menu item unselects all selected archive entries in the Archive Media window.

4.5.4.4 View Archive Database Window

The View Archive Database window is opened when you choose **View Archive Database** from the **Tools** menu on the History Builder menu bar.

4.5.4.5 View Archive Database - File Menu

The **File** menu for the View Archive Database window is shown in Figure 4-62.

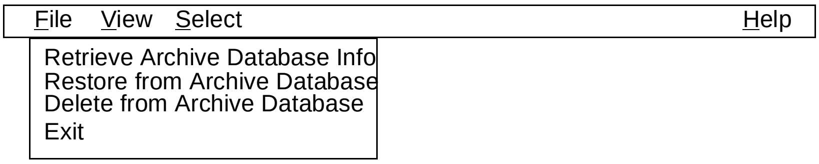
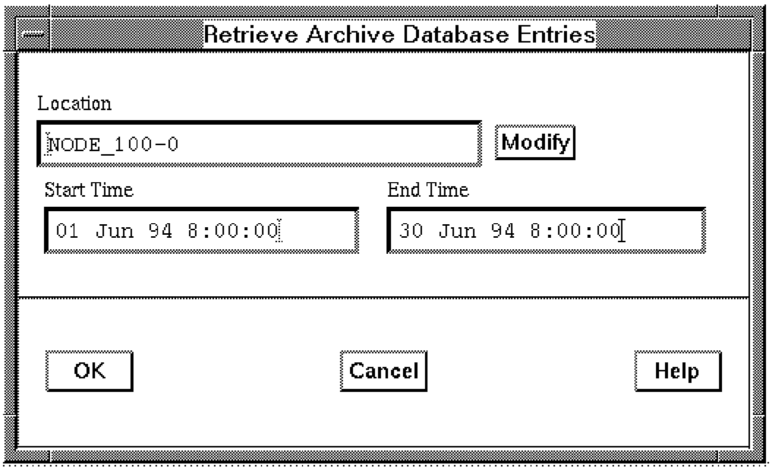

File
View
Select
Help
Retrieve Archive Database Info
Restore from Archive Database
Delete from Archive Database
Exit

Figure 4-62. Archive Database - File Menu

File - Retrieve Archive Database Info

Retrieve Archive Database Info is used to retrieve and view archive database entries. This menu item opens the Retrieve Archive Database window, Figure 4-63.



Location
NODE_100-0

Start Time
01 Jun 94 8:00:00

End Time
30 Jun 94 8:00:00

Figure 4-63. Retrieve Archive Data Dialog

Location	Node where you are retrieving archive database entries from.
Start Time & End Time	Start and end times for archive database entries. The format for start and end times is as follows: day of the month, month (first three letters), year (last two digits), hours:minutes:seconds. For example: `07 Jun 94 09:35:00`.

File - Restore from Archive Database

Restore from Archive Database is used to restore log entries. This menu item opens the Restore Archive DB dialog, Figure 4-64.

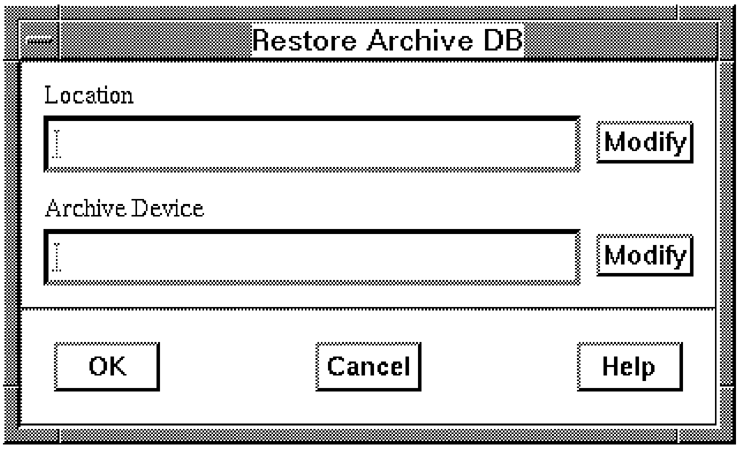


Figure 4-64. Restore Archive DB Dialog

Location	Node where you are restoring archive database entries from.
Archive Device	Archive device to restore from.

File -Delete from Archive Database

Delete from Archive Database is used to delete selected archived logs from the History database.

File - Exit

Exit is used to close the Archive Database window.

4.5.4.6 Archive Database Window - View Menu

The **View** menu for the Archive Database window is shown in Figure 4-65.

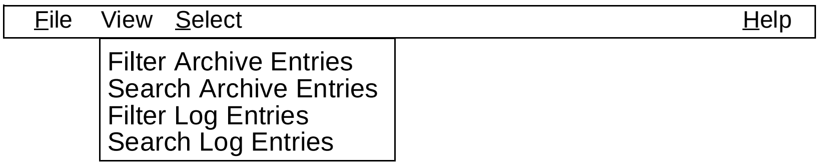


Figure 4-65. View Menu Items

View - Filter Archive Entries

Filter Archive Entries is used to reduce the number of archive entries displayed by applying a filter based on a user-defined combination of volume ID, archive group and archive type. When you choose **Filter Archive Entries**, the Filter Archive Entries dialog box is displayed, Figure 4-57.

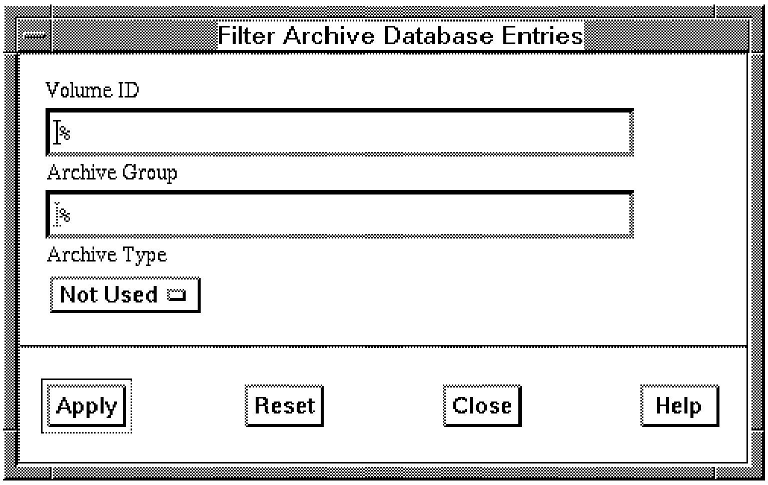


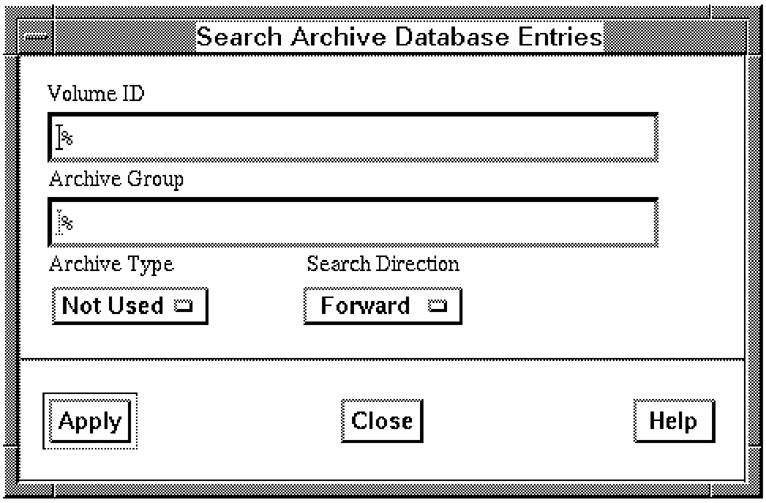
Figure 4-66. Filter Archive Database Entries Dialog

The filter parameters are described below. You can use these parameters in any combination as required.

Volume ID	Only display archive entries whose volume ID is specified in this field. % is a wild card text string.
Archive Group	Only display archive entries whose archive group is specified in this field. % is a wild card text string.
Archive Type	Only display archive entries whose archive type is as specified in this field (MANUAL or TIMED).

View - Search Archive Entries

Search Archive Entries is used to search for a specific archive entry. This function finds the first archive entry in the displayed list that matches the specified parameters, and scrolls the list so that archive entry is at the top. When you choose **Search Archive Entries**, the Search Archive Database dialog box is displayed, Figure 4-58. The parameters are described below.



Volume ID: %

Archive Group: %

Archive Type: Not Used

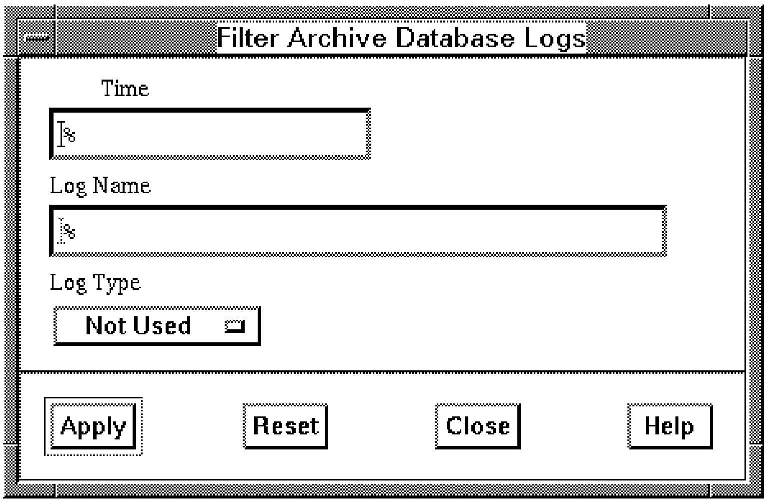
Search Direction: Forward

Figure 4-67. Search Archive Database Dialog

Volume ID	Search for archive entries whose volume ID is specified in this field. % is a wild card text string.
Archive Group	Search for archive entries that belong to specified archive group. % is a wild card text string.
Archive Type	Search for archive entries of specified archive type (MANUAL or TIMED).
Search Direction	Displays the selected search direction (forward or backward). To change the search direction, click on the right hand button.

View - Filter Log Entries

Filter Log Entries is used to reduce the number of archive entries displayed by applying a filter based on a user-defined combination of log name, log type, and time range. When you choose **Filter Log Entries**, the Filter Archive Database Logs dialog box is displayed, Figure 4-57.



Time: %

Log Name: %

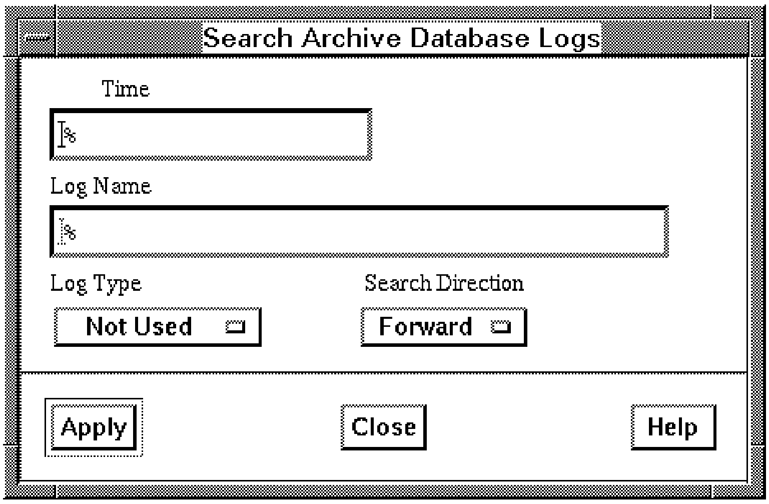
Log Type: Not Used

Figure 4-68. Filter Archive Database Logs Dialog

Log Name	Only display logs whose name has the specified text string. % is a wild card text string.
Log Type	Only display logs whose log type is as specified in this field.
Time	Only display logs whose start and end times cover the time specified in this field. For example, if the log start time is 30 aug 95 10:00:00 and the end time is 30 aug 95 12:00:00, and the filter time entered in this dialog is 30 aug 95 10:45:00, the log will be displayed. The format for time is as follows: day of the month, month (first three letters), year (last two digits), hours:minutes:seconds. For example: `07 Jun 95 09:35:00`.

View - Search Log Entries

Search Archive Entries is used to search for a specific archive entry. This function finds the first archive entry in the displayed list that matches the specified parameters, and scrolls the list so that archive entry is at the top. When you choose **Search Archive Entries**, the Search Archive Database Logs dialog box is displayed, Figure 4-58. The parameters are described below.



Time: %

Log Name: %

Log Type: Not Used

Search Direction: Forward

Figure 4-69. Search Archive Database Logs Dialog

Log Name	Search for logs whose name has the specified text string. % is a wild card text string.
Log Type	Search for logs whose log type is as specified in this field.
Time	Only display logs whose start and end times cover the time specified in this field. For example, if the log start time is 30 aug 95 10:00:00 and the end time is 30 aug 95 12:00:00, and the filter time entered in this dialog is 30 aug 95 10:45:00, the log will be displayed. The format for time is as follows: day of the month, month (first three letters), year (last two digits), hours:minutes:seconds. For example: `07 Jun 95 09:35:00`.
Search Direction	Displays the selected search direction (forward or backward). To change the search direction, click on the right hand button.

4.5.4.7 Archive Database Window - Select Menu

The **Select** menu for the Archive Database window is shown in Figure 4-61.

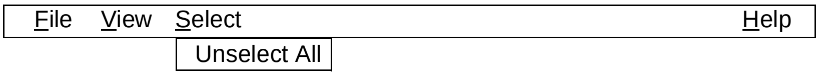


Figure 4-70. Select Menu

Select - Unselect All

This menu item unselects all selected archive entries in the Archive Database window.

4.5.5 Menus for the PDL Window (MOD 300)

4.5.5.1 PDL Window - File menu

File Delete PDLs

This menu item deletes the PDL or PDLs currently selected in the PDL window. A confirmation message box is displayed so you have the opportunity to confirm whether or not you actually want to delete the PDL. When you delete a PDL, all batch, unit procedure, and phase tasks under that PDL are deleted, as are all log and resource associations that belong to that job or batch.The object is deleted from History.

File Delete Tasks

This menu item deletes one or more batch-level tasks under the PDL currently selected in the PDL window. A confirmation message box is displayed so you have the opportunity to confirm whether or not you actually want to delete the task(s). When you delete a batch task, all unit procedure, and phase tasks under that batch are also deleted.

Exit

This menu item closes the PDL window.

4.5.6 Menus for the Runtime History Status Window

4.5.6.1 Runtime History Status, Detail Display - File menu

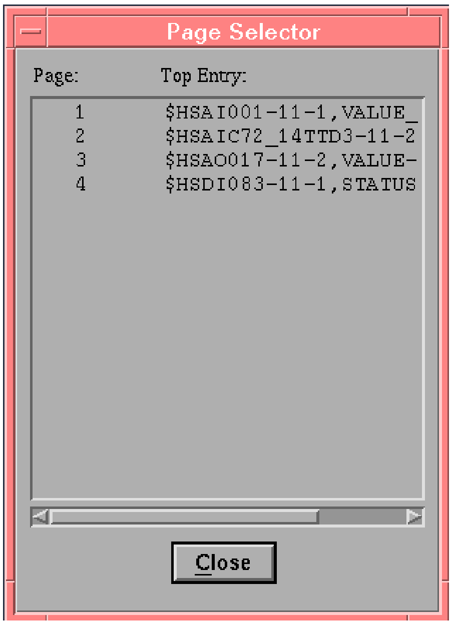
Exit

Use this menu item to exit the current list displayed in the Runtime History Status Window.

4.5.6.2 Runtime History Status, Detail Display - Tools menu

Log Index

This menu item displays the Page Selector dialog. This dialog provides a list of pages for the current list, Figure 4-71. You can navigate directly to a page in a list by selecting the page from this dialog.



Page:	Top Entry:
1	$HSAI001-11-1,VALUE_
2	$HSAIC72_14TTD3-11-2
3	$HSAO017-11-2,VALUE-
4	$HSDI083-11-1,STATUS

Figure 4-71. Page Selector

Open Custom List

This menu item displays the Custom Log List dialog for building a custom log list. Refer to Section 4.4.1.2, How to Build a Custom List for instructions on building a custom list.

Configure

This menu item lets you change the size of a log status page (change the number of logs per page, and thus, the total number of pages). Refer to Section 4.4.1.7, How to Change Which Attributes are Displayed in the Detail Window.

Clear messages

This menu item clears all messages from the Messages field of the Detail Display.

Format Attribute List

This menu item lets you specify which attributes to show and which attributes to hide in the Attributes part of the Detail Display. Refer to Section 4.4.1.7, How to Change Which Attributes are Displayed in the Detail Window.

4.5.6.3 Runtime History Status Detail Display - Actions Menu

Activate

This menu item activates the selected log(s). First select the log(s) that you want to activate. To select individual non-contiguous logs, press and hold the **Ctrl** key as you select the logs. To select a group of contiguous logs, select the first log, then press and hold the **Shift** key as you select the last log in the contiguous group. To un-select an individual log, press and hold the **Ctrl** key as you click on the selected the log. Then choose **Activate** from the **Actions** menu.

Deactivate

This menu item deactivates the selected log(s). First select the log(s) that you want to deactivate. To select individual non-contiguous logs, press and hold the **Ctrl** key as you select the logs. To select a group of contiguous logs, select the first log, then press and hold the **Shift** key as you select the last log in the contiguous group. To un-select an individual log, press and hold the **Ctrl** key as you click on the selected the log. Then choose **Deactivate** from the **Actions** menu.

Deselect All

This menu item deselects all currently selected logs. Selected logs are shown highlighted. When you deselect a log, the log is no longer highlighted.

Add to Custom List

This menu item adds all selected logs to a custom list. Refer to Section 4.4.1.2, How to Build a Custom List for details.

Activate All

This menu item activates all logs in the list, regardless of which logs are selected and which logs are deselected.

Deactivate All

This menu item deactivates all logs in the list, regardless of which logs are selected and which logs are deselected.

Refresh List

This menu item is equivalent to the Refresh button on the Detail Display.

Chapter 5 Maintenance

5.1 Preventive Maintenance

5.1.1 Problem Reporting

If a problem occurs that can not be solved by you yourself or your system manager, please contact your local ABB Industrial Systems representative. This includes problems and errors that you have identified as being related to the AdvaInform Basic Software, AdvaInform and AdvaBuild options, the operating system HP-UX, Oracle RDBMS or any Oracle tool.

When you report a problem, please enclose all of the supplementary information related to the problem that you have. It might be for example a description of changes you made in your system prior to the problem. It might also be printouts of error logs and hard copies of, for example, configuration displays. The more information you can supply the shorter time for a response.

5.1.2 The HP-UX kernel

You should avoid to change the settings of the HP-UX kernel. Buffers, swap space, semaphores and so on have been set to insure that Advant Station has the capacity to handle AdvaInform functionality and still keep the best possible performance.

Some third party software might require rebuild of the HP-UX kernel. The system manager may only increase a system parameter. Note that you in that case change the capacity and performance of all functions in Advant Station.

5.2 Hardware Indicators

Not Applicable

5.3 Error Messages for Systems with Master Software

The error messages generated by the AdvaInform Basic Software and options (AdvaInform and AdvaBuild) can be viewed in different ways - for example only those having a certain severity code - by using the features of the System Message Display (described in Chapter 4 of the *AdvaInform Basic Functions User's Guide*).

The error messages issued by the AdvaInform Basic Software and options have the following standardized format:

Error string <P1><P2><P3>

Parameters: P1: description of parameter 1
P2: description of parameter 2
P3: description of parameter 3
The parameters are optional in an error message. This means that there might be no parameter, just a text, or one, two or three parameters.

Audience: The audience of the error message. Valid entries are Operator, Engineer, Service and Debug.

Cause: Cause of the error.

Action: Action to take when the error occurs.

The error messages are sorted after their severity in the following classes

- Fatal error messages
- Error messages
- Warning messages
- Information messages

5.3.1 Oracle *oerr* on-line tool

Error messages related to *Oracle* tools can be further investigated by using the *oerr* on-line tool from a terminal window - provided your user category is Engineer or System. This applies to all of the error messages issued by any of the installed Oracle tools. This type of messages may appear in two ways:

- In interactive mode (for example when using the Object Forms) the message (starting with the ORA number) is displayed at the bottom line of the window
- In a *trace* file. This occurs when a failure in hardware or software (system or an Oracle tool) prevents an instance from continuing work.

The trace files can be found in the directory `/var/opt/advant/log/oracle`. The files are in ascii format. To find the trace file of interest for your problem, check the date of the files. Type the file to get the error number and use *oerr* to get a description.

Example

Suppose you want to get a closer description of the Oracle message ORA-6100 than what is supplied by the message itself:

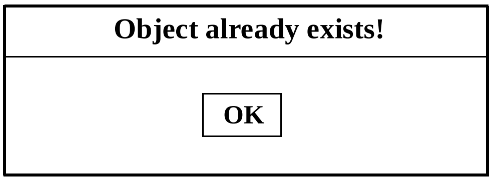
- Open a terminal window (can be done from the IMS menu bar, see Chapter 4 in the *AdvaInform Basic Functions User’s Guide*)
- Enter command: *oerr ora 6100*

This will give you a description of the *cause* of the error as well as a suggestion on what *action* you should take to cure the problem.

5.3.2 Messages Box

The other type of error (or informational) messages that are issued by the AdvaInform Basic Software, AdvaInform (History in this case) and AdvaBuild options are presented directly to the operator. Normally, the reason for the message is that the operator has entered an invalid command. In most cases, the operator himself can take the necessary actions to correct the command.

This type of messages are presented in a standardized message box with the following appearance:



Object already exists!

Figure 5-1. Example of an Error Message Dialogue Box

The error message must be acknowledged (click on OK) before you are allowed to continue.

The error messages are self explanatory and are thus not further described here.

5.3.3 History Diagnostic Messages

5.3.3.1 Main History

HISTORY <P1> ACTIVATED

Parameters: P1: Name of the function activated (may be empty string)

Audience: Operator

Cause: Information message.

Action:

HISTORY <P1> DEACTIVATED

Parameters: P1: Name of the function deactivated (may be empty string)

Audience: Operator

Cause: Information message.

Action:

HISTORY SHUT DOWN

Parameters: -

Audience: Operator

Cause: Information message.

Action:

HISTORY INITIALIZATION FAILED

Parameters: -

Audience: Operator

Cause: May have a number of reasons.

Action: Contact ABB representative.

OPERATION OR ATTRIBUTE <P1> (NUMBER <P2>) NOT IMPLEMENTED

Parameters: P1: Name of the operation to attribute
P2: Operation or attribute Id

Audience: Operator

Cause: Functionality for the operation or attribute is not yet implemented

Action:

<P1> OPERATION FAILED ON LOG ID <P2>

Parameters: P1: Operation name

P2: Operation Id

Audience: Operator

Cause: System failure, - refer to lower level service/engineer diagnostics. Operation was not performed.

Action: Verify that the log is properly configured.

UNABLE TO INITIALIZE LOG STATES <P1>

Parameters: P1: OTI (Oracle) error code

Audience: Operator

Cause: Unable to update the states (Oracle error). This means that the state the log has in the database will not match the actual state.

Action: Contact ABB representative.

5.4 Error Messages for Systems with MOD 300 Software

5.4.1 Accessing Error Messages

Throughout the use of this application, various informational and diagnostic messages are generated. To access a listing of these messages, select the DIAG target in the upper right corner of your display. This action automatically takes you to the DIAGNOSTIC MESSAGES DISPLAY and presents a sentence or phrase to help explain the situation or error you have encountered.

However, on occasion, you may require outside assistance for the successful resolution of a situation or problem. If this is the case, the diagnostic error messages contained in this section may help.

If you need to contact service for help, it is important to provide them with as much information as possible concerning your problem. To obtain more information, you must change the way the diagnostic messages are displayed. Select the OPERATOR_MSGS target from the DIAGNOSTIC MESSAGES DISPLAY. Using the NEXT CHOICE key, toggle to and then select ALL MSGS—CODE. This will provide you with the AP code associated with the error that has occurred. Find the AP code within this section to determine the cause and effect of your problem as well as the action you should take next. Have this information on hand if you need to contact the service organization for additional support.

NOTE

Please record all messages that are not found in this manual and report them to ABB. The information you record should contain the following:

- error message text copied exactly as it appears on the screen

- error message code copied exactly as it appears on the screen
- type of node that generated the diagnostic
- sequence of events leading up to the diagnostic message
- any other important data

Providing the above information allows for a more efficient response and accurate documentation.

5.4.2 Error Message Format

The diagnostic error messages are presented in the following format:

AP Error Code (Severity)	**Message ID:**
	Message Text:
	Data 1 (%D1):
	Data 2 (%D2):
	Text Data (%T2):
	Cause:
	Effect:
	Audience:

AP Error Code	Four digit hexadecimal code. The first two digits (AP) are the application ID, the last two digits are the message number.
Severity	Information (I),Warning (W), Error (E), Fatal (F)
Message ID	Text message ID
Message Text	Displayed when ALL_MESSAGES_TEXT is selected on the diagnostic display.
Data 1 (%D1)	Numerical data 1
Data 2 (%D2)	Numerical data 2
Text Data (%T)	Message specific text
Cause	Reason for message
Effect	Effect of condition on system
Audience	Intended Audience (Operator, Engineer, Service, Debug)

5.4.3 History Diagnostic Messages

The history messages are organized in sections in this chapter. Each section is associated with a particular series of error codes. For example, Section 5.3.3.1, contains all error codes associated with the AP code 81 (Main History). Therefore, all error codes associated with the Main History application will be found in Section 5.4.3.1, AP 81 - Main History.

5.4.3.1 AP 81 - Main History

8101 W	**Message ID:**	hsBSSYS
	Message Text:	SYSTEM ERROR %D1 RETURNED FROM SERVICE %T
	Data 1 (%D1):	Errno
	Data 2 (%D2):	
	Text Data (%T):	Service name
	Cause:	
	Effect:	
	Audience:	SERVICE

8102 W **Message ID:** hsBSOMF
Message Text: OMF ERROR %D1 RETURNED FROM SERVICE %T
Data 1 (%D1): OMF status
Data 2 (%D2):
Text Data (%T): OMF function name
Cause:
Effect:
Audience: SERVICE

8103 W **Message ID:** hsBSOTI
Message Text: OTI ERROR %D1 FROM TRANS ON %T (%D2)
Data 1 (%D1): OTI status
Data 2 (%D2): Table Id
Text Data (%T): Table name
Cause:
Effect:
Audience: SERVICE

8104 W **Message ID:** hsBSIPC
Message Text: IPC ERROR %D1 RETURNED FROM SERVICE %T
Data 1 (%D1): IPC status
Data 2 (%D2):
Text Data (%T): IPC service name
Cause:
Effect:
Audience: SERVICE

8105 W **Message ID:** hsBSRTS
Message Text: RTS ERROR %D1 RETURNED FROM SERVICE %T
Data 1 (%D1): RTS status
Data 2 (%D2):
Text Data (%T): RTS service name
Cause:
Effect:
Audience: SERVICE

8106 I **Message ID:** hsBSACTIV
Message Text: HISTORY %T ACTIVATED
Data 1 (%D1):
Data 2 (%D2):
Text Data (%T): Name of the function activated (may be empty string)
Cause:
Effect:
Audience: OPERATOR

8107 W **Message ID:** hsBSDEACTIV
Message Text: HISTORY %T DEACTIVATED
Data 1 (%D1):
Data 2 (%D2):
Text Data (%T): Name of the function deactivated (may be empty string)
Cause:
Effect:

Audience: OPERATOR

8108 W **Message ID:** hsBSSHUT_DOWN
Message Text: HISTORY SHUT DOWN
Data 1 (%D1):
Data 2 (%D2):
Text Data (%T):
Cause:
Effect:
Audience: OPERATOR

8109 E **Message ID:** hsBSINIT_FAIL
Message Text: HISTORY INITIALIZATION FAILED
Data 1 (%D1):
Data 2 (%D2):
Text Data (%T):
Cause:
Effect:
Audience: OPERATOR

8110 E **Message ID:** hsBSRTRV_FAIL
Message Text: RETRIEVAL OF OBJECT DATA FAILED
Data 1 (%D1):
Data 2 (%D2):
Text Data (%T):
Cause:
Effect:
Audience: ENGINEER

8111 E **Message ID:** hsBSMDFY_FAIL
Message Text: MODIFICATION OF OBJECT DATA FAILED
Data 1 (%D1):
Data 2 (%D2):
Text Data (%T):
Cause:
Effect:
Audience: ENGINEER

8112 E **Message ID:** hsBSADD_FAIL
Message Text: ADDITION OF OBJECT DATA FAILED
Data 1 (%D1):
Data 2 (%D2):
Text Data (%T):
Cause:
Effect:
Audience: ENGINEER

8113 E **Message ID:** hsBSCRT_FAIL
Message Text: CREATION OF OBJECT FAILED
Data 1 (%D1):
Data 2 (%D2):
Text Data (%T):

Cause:
Effect:
Audience: ENGINEER

8114 E **Message ID:** hsBSINTERNAL
Message Text: INTERNAL INCONSISTENCY %T
Data 1 (%D1):
Data 2 (%D2):
Text Data (%T): Logical expression
Cause:
Effect:
Audience: SERVICE

8115 E **Message ID:** hsBSNOT_IMPLEMENTED
Message Text: OPERATION OR ATTRIBUTE %T (NUMBER %D1) NOT IMPLEMENTED
Data 1 (%D1): Operation or attribute Id
Data 2 (%D2):
Text Data (%T): Name of the operation to attribute
Cause: Functionality for the operation or attribute is not yet implemented
Effect:
Audience: OPERATOR

8116 E **Message ID:** hsBSATTRIBUTE_TYPE
Message Text: INVALID ATTRIBUTE (%D1) DATA TYPE %D2
Data 1 (%D1): History attribute Id
Data 2 (%D2): Invalid attribute type
Text Data (%T):
Cause: The data type that was found for an attribute was invalid
Effect:
Audience: SERVICE

8117 E **Message ID:** hsBSCLASS_ID
Message Text: INVALID OBJECT CLASS ID %D1
Data 1 (%D1): Class Id
Data 2 (%D2):
Text Data (%T):
Cause: Unknown class Id (object type)
Effect:
Audience: SERVICE

8118 E **Message ID:** hsBSDB_CONNECT
Message Text: %T UNABLE TO SUCCESSFULLY CONNECT TO DATABASE
Data 1 (%D1):
Data 2 (%D2):
Text Data (%T): Application name
Cause: Error connecting to the database
Effect:
Audience: SERVICE

8119 W **Message ID:** hsBSMEMORY
Message Text: UNABLE TO ALLOCATE MEMORY (%D1) (%T)
Data 1 (%D1): Amount of memory to allocate

Data 2 (%D2):
Text Data (%T): Name of call
Cause: Unable to allocate needed memory
Effect:
Audience: SERVICE

8120 I
Message ID: hsBSLOGSTATE
Message Text: LOG ID (%D1) (%T)
Data 1 (%D1): History data base log Id
Data 2 (%D2):
Text Data (%T): Description of state (just activated, already inactive, ...)
Cause: Attempting to activate, deactivate or add to a log
Effect:
Audience: SERVICE

8121 W
Message ID: hsBSLOGTYPE
Message Text: INVALID LOG TYPE (%D1)
Data 1 (%D1): Log type
Data 2 (%D2): Number of bytes attempting to allocate
Text Data (%T):
Cause: Invalid log type encountered
Effect:
Audience: SERVICE

8122 W
Message ID: hsBSCOLLECTION
Message Text: LOG ID (%D1), DATA/STATUS (%D2) FROM (%T)
Data 1 (%D1): History data base log Id
Data 2 (%D2): Data specific to message
Text Data (%T): Name of subroutine called or brief description
Cause: Miscellaneous
Effect:
Audience: SERVICE

8123 E
Message ID: hsBSHLDMINIT
Message Text: HLDM INIT FAILED WITH STATUS %D1
Data 1 (%D1): HLDM status
Data 2 (%D2):
Text Data (%T):
Cause:
Effect:
Audience: SERVICE

8124W
Message ID: hsBSNOATTRSRET
Message Text: NONE OF THE REQUESTED ATTRIBUTES WERE RETURNED
Data 1 (%D1):
Data 2 (%D2):
Text Data (%T):
Cause: The requesting client asked for XX attributes and none were found
Effect:
Audience: ENGINEER

8125 E **Message ID:** hsBSOP_FAILED
Message Text: %T OPERATION FAILED ON LOG ID %D1
Data 1 (%D1): Operation Id
Data 2 (%D2): Log Id, if a log object
Text Data (%T): Operation name
Cause: System failure -- refer to lower level service/engineer diagnostics
Effect: Operation was not performed
Audience: OPERATOR

8126 E **Message ID:** hsBSLOG_START_STATE
Message Text: UNABLE TO INITALIZE LOG STATES %D
Data 1 (%D1): OTI (Oracle) error code
Data 2 (%D2):
Text Data (%T):
Cause: Unable to update the states. (Oracle error)
Effect: The state the log has in the database will not match the actual state
Audience: OPERATOR

5.4.3.2 AP 82 - Log Manager

8201 W **Message ID:** hsBSSYS
Message Text: SYSTEM ERROR %D1 RETURNED FROM SERVICE %T
Data 1 (%D1): Errno
Data 2 (%D2):
Text Data (%T): Service name
Cause:
Effect:
Audience: SERVICE

8202 W **Message ID:** hsBSOMF
Message Text: OMF ERROR %D1 RETURNED FROM SERVICE %T
Data 1 (%D1): OMF status
Data 2 (%D2):
Text Data (%T): OMF function name
Cause:
Effect:
Audience: SERVICE

8203 I **Message ID:** hsBSACTIV
Message Text: HISTORY %T ACTIVATED
Data 1 (%D1):
Data 2 (%D2):
Text Data (%T): Name of the function activated (may be empty string)
Cause:
Effect:
Audience: OPERATOR

8204 W **Message ID:** hsBSSHUT_DOWN
Message Text: HISTORY SHUT DOWN
Data 1 (%D1):
Data 2 (%D2):

Text Data (%T):
Cause:
Effect:
Audience: OPERATOR

8205 E **Message ID:** hsBSINIT_FAIL
Message Text: HISTORY INITIALIZATION FAILED
Data 1 (%D1):
Data 2 (%D2):
Text Data (%T):
Cause:
Effect:
Audience: OPERATOR

8206 E **Message ID:** hsBSNOT_IMPLEMENTED
Message Text: OPERATION OR ATTRIBUTE %T (NUMBER %D1) NOT IMPLEMENTED
Data 1 (%D1): Operation or attribute Id
Data 2 (%D2):
Text Data (%T): Name of the operation to attribute
Cause: Functionality for the operation or attribute is not yet implemented
Effect:
Audience: OPERATOR

8207 W **Message ID:** hsBSMEMORY
Message Text: UNABLE TO ALLOCATE MEMORY (%D1) (%T)
Data 1 (%D1): Amount of memory to allocate
Data 2 (%D2):
Text Data (%T): Name of call
Cause: Unable to allocate needed memory
Effect:
Audience: SERVICE

5.4.3.3 AP 83 - History Builder

8301 W **Message ID:** hsBSSYS
Message Text: SYSTEM ERROR %D1 RETURNED FROM SERVICE %T
Data 1 (%D1): Errno
Data 2 (%D2):
Text Data (%T): Service name
Cause:
Effect:
Audience: SERVICE

8302 W **Message ID:** hsBSOMF
Message Text: OMF ERROR %D1 RETURNED FROM SERVICE %T
Data 1 (%D1): OMF status
Data 2 (%D2):
Text Data (%T): OMF function name
Cause:
Effect:
Audience: SERVICE

8303 I **Message ID:** hsBSACTIV
Message Text: HISTORY %T ACTIVATED
Data 1 (%D1):
Data 2 (%D2):
Text Data (%T): Name of the function activated (may be empty string)
Cause:
Effect:
Audience: OPERATOR

8304 W **Message ID:** hsBSSHUT_DOWN
Message Text: HISTORY SHUT DOWN
Data 1 (%D1):
Data 2 (%D2):
Text Data (%T):
Cause:
Effect:
Audience: OPERATOR

8305 E **Message ID:** hsBSINIT_FAIL
Message Text: HISTORY INITIALIZATION FAILED
Data 1 (%D1):
Data 2 (%D2):
Text Data (%T):
Cause:
Effect:
Audience: OPERATOR

8306 W **Message ID:** hsBSMEMORY
Message Text: UNABLE TO ALLOCATE MEMORY (%D1) (%T)
Data 1 (%D1): Amount of memory to allocate
Data 2 (%D2):
Text Data (%T): Name of call
Cause: Unable to allocate needed memory
Effect:
Audience: SERVICE

5.5 Fault Finding and Repair

The following tools are available to help you find the reason for a malfunction in your Advant Station:

- The message on your screen

  Normally, the error message provides enough information for you to find and fix the problem. For messages from Oracle tools the *oerr* on-line tool is also available.

- The System Message Display (for Advant Station w/AdvaInform software only)

  Details regarding History error messages are provided in this chapter. The *AdvaInform Basic Functions User's Guide* describes how to use the System Message display.

- The Oracle trace files (for Advant Station w/AdvaInform software only)

The information in those files together with the *oerr* on-line tool will help you. In Section 5.3.1, Oracle oerr on-line tool you can read where to find the trace files and how to use the tool.

- The Status displays

  In Chapter 4 you can find information about what status displays that are available and what information you can find in them. For example, you can check whether a (History) process related to your problem is running or not.

- Operating system services

  Of course you can also use the tools provided by the operating system you are running to get, for example, information about different processes.

NOTE

If the Advant Station is stopped in an uncontrolled manner, the relational database may crash and get corrupted. Therefore, stop Oracle before you shut down the node or reboot. Instructions are provided in the *Advant Station 500 Series IMS User's Guide*.

Before you start extensive trouble-shooting procedures, make sure there is no “simple” reason for your problem. For example, make sure that you really are authorized to perform the function you are trying to do. Make sure there is nothing in your configuration data that spoils the function and so on.

5.6 Database Maintenance Functions

Database maintenance functions are provided via the hsDBMaint command which resides in the $HS_HOME/bin directory. To use the hsDBMaint functions, you must be logged into AdvaInform as the ocsmgr user, or into AdvaCommand as operator.

The database maintenance functions provided by hsDBMaint are:

- Database Free Space report

  Use this function to generate a report on free space in Oracle tablespaces on the History disk.

- Entry Tables Report

  Use this function to create a comprehensive list of all numeric logs, their current capacity, and other useful statistics.

- Update Deadband Ratio On-line

  Use this function to update deadband ratio and estimated log time period for logs that use deadband.

- Create and drop a product database

  Use this function to create a new data base of type History, PDL, or SPC, or to drop (delete an existing database).

- Extend a Tablespace

  Use this function to extend the tablespace in your system when you require greater storage capacity than is provided by the default Oracle tablespace sizes.

- Add, alter, delete, and list Numeric log directories

  Use this function to configure directories for file-based storage. This is the same functionality provided by the Directory Maintenance window.

- Reset object status states

  Use this function to reset *attrsUpdated* and *userEntered* bits. These bits determine whether the collection function will update the trend presentation attributes, or whether user-entered values will be used.

- Cascade Attributes for Composite Logs

  Use this function to propagate presentation, engineering units, and alarm attributes from the data source to secondary logs in the composite hierarchy (see Section 3.1.5, Presentation (Trend), Engineering Units, and Alarm Attributes). These attributes can be read directly from the data source to the primary log, but must be passed to secondary logs via this function. In order for deadband to work properly on the secondary log, the node must be re-started after the cascade is done.

- Gather performance statistics

  Use this function to gather CPU usage data to analyze performance.

- Purge History Data

  Use this function to clear History data that was erroneously collected in the future.

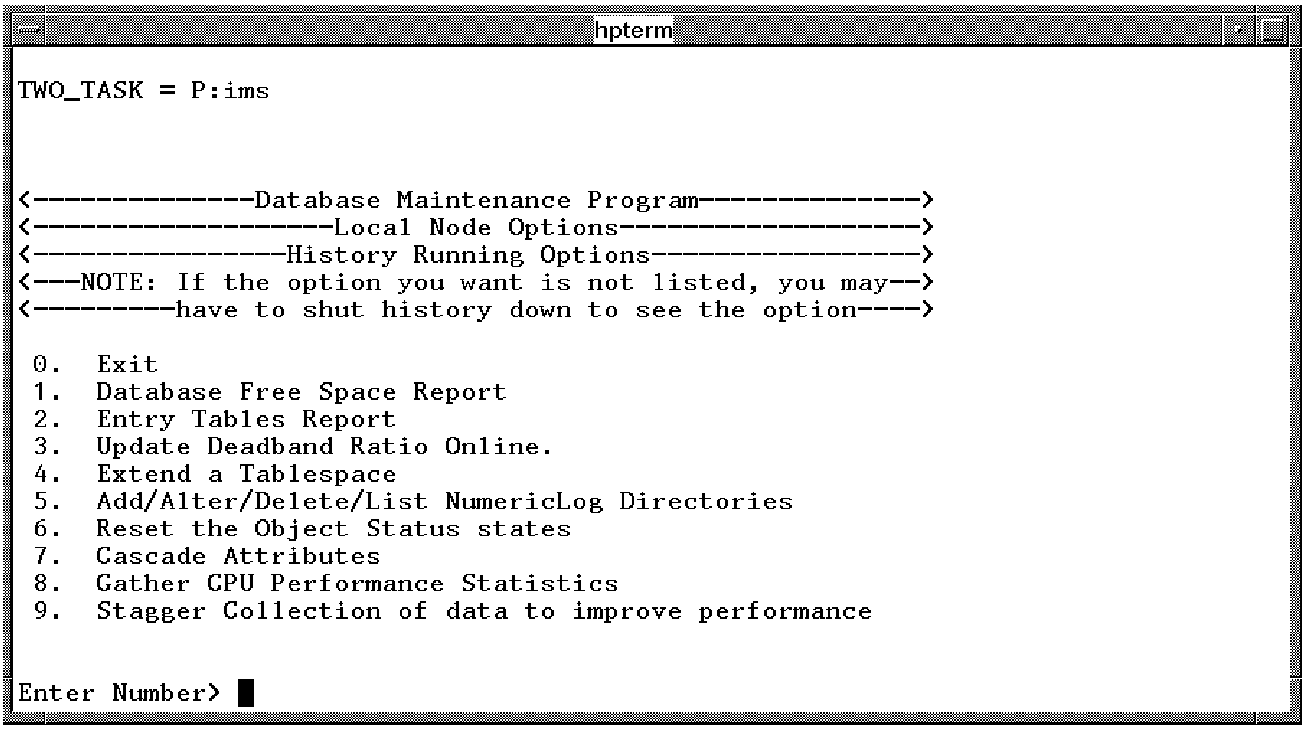
- Clean History Database

  Use this function to clean up the History database. In the course of creating and deleting logs, if an operation is interrupted for some reason, it may leave what are referred to as dangling references in the database. For instance, if the creation of a log is not completed for some reason, the log name is used in the database even though the log does not actually exist. This means the log name can not be re-used. The cleanup function deletes these dangling references from the database. In the case of the dangling log name, this means that the name can be used again.

- Restore and initialize history logs

5.6.1 How to Access hsDBMaint Functions

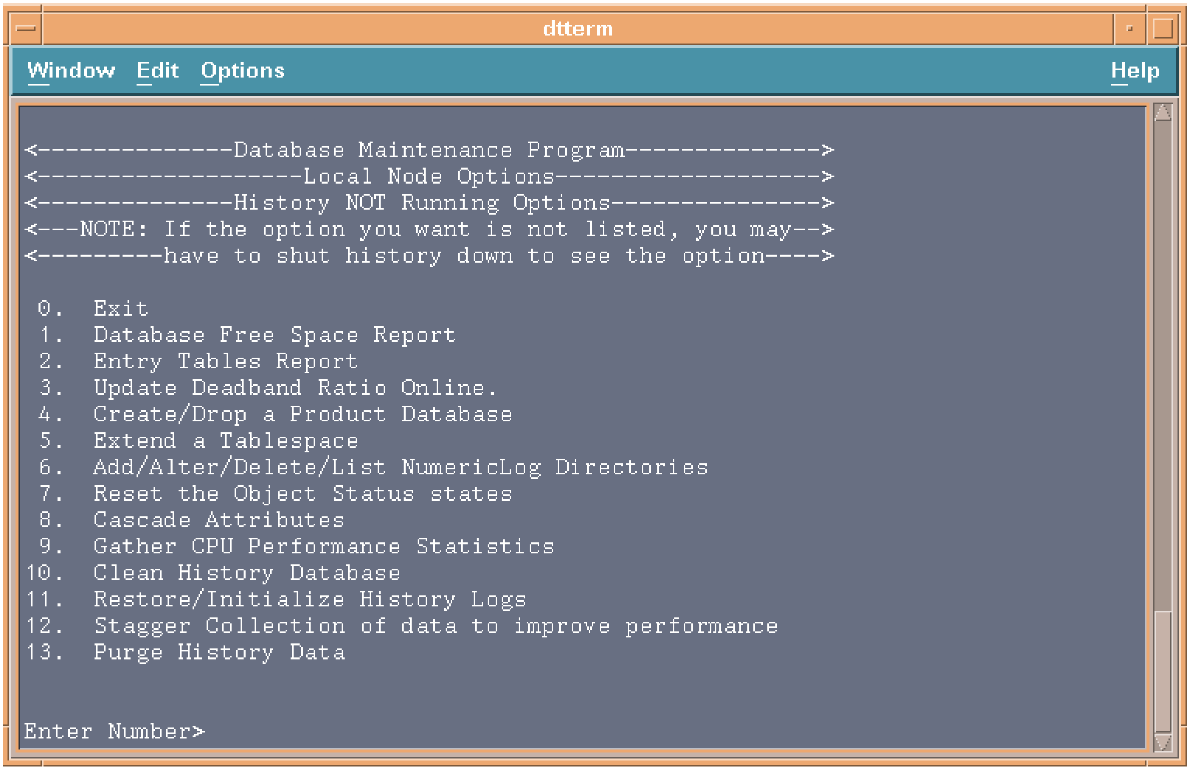
The hsDBMaint functions are available in the Database Maintenance Program Main Menu. They can also be invoked from the command line. Not all hsDBMaint functions are available when History is running. This is to prevent you from performing certain maintenance functions that can not be allowed when history is running. The menu for on-line History is shown in Figure 5-2.



```
TWO_TASK = P:ims



<--------------Database Maintenance Program-------------->
<-------------------Local Node Options------------------->
<----------------History Running Options---------------->
<---NOTE: If the option you want is not listed, you may-->
<---------have to shut history down to see the option---->

 0.   Exit
 1.   Database Free Space Report
 2.   Entry Tables Report
 3.   Update Deadband Ratio Online.
 4.   Extend a Tablespace
 5.   Add/Alter/Delete/List NumericLog Directories
 6.   Reset the Object Status states
 7.   Cascade Attributes
 8.   Gather CPU Performance Statistics
 9.   Stagger Collection of data to improve performance


Enter Number> █
```

Figure 5-2. Database Maintenance Menu when History is Running

The menu for off-line History is shown in Figure 5-3.



```
<--------------Database Maintenance Program-------------->
<------------------Local Node Options------------------->
<--------------History NOT Running Options--------------->
<---NOTE: If the option you want is not listed, you may-->
<---------have to shut history down to see the option---->

 0.  Exit
 1.  Database Free Space Report
 2.  Entry Tables Report
 3.  Update Deadband Ratio Online.
 4.  Create/Drop a Product Database
 5.  Extend a Tablespace
 6.  Add/Alter/Delete/List NumericLog Directories
 7.  Reset the Object Status states
 8.  Cascade Attributes
 9.  Gather CPU Performance Statistics
10.  Clean History Database
11.  Restore/Initialize History Logs
12.  Stagger Collection of data to improve performance
13.  Purge History Data


Enter Number>
```

Figure 5-3. Database Maintenance Menu when History is off-line

NOTE

If you need to perform an off-line maintenance function, you must first stop Advant Station processes (if they are not already stopped). If you are running History with AdvaInform, use the **`IMSstop`** command as described in the *Advant Station 500 Series IMS User's Guide*. If you are running History with AdvaCommand, use the **`quit`** command as described in the *Advant Station Operator Station User's Guide*.

To access hsDBMaint functions:

1. Log in to AdvaInform as ocsmgr or AdvaCommand as operator user.
2. Open a terminal window and enter the following commands:

   ```
   $cd $HS_HOME/bin
   $ hsDBMaint
   ```

   This displays the Database Maintenance Program Main Menu (hereafter referred to as Main Menu).
3. To invoke a function, enter the corresponding number, and then press <**Return**>.

For example, to create a Database Free Space report, enter **1** and then press <**Return**>.

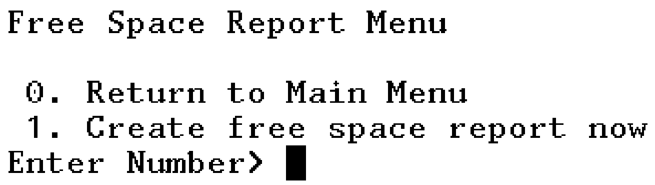
4. Then follow the instructions for the selected function. For example, Section 5.6.3, Database Free Space Report.

5.6.2 Exit hsDBMaint

To exit the hsDBMaint function, choose **Exit** from the Main Menu.

5.6.3 Database Free Space Report

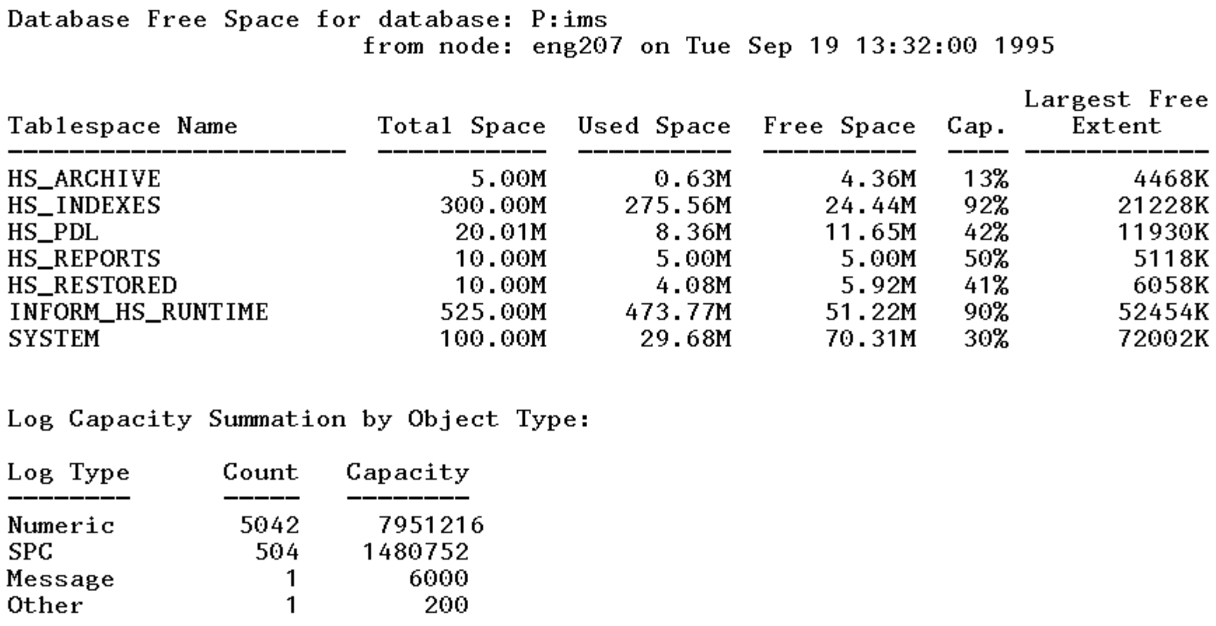
The Free Space Report indicates how much space is available in the tablespaces. To create a Free Space Report, choose **Database Free Space Report** from the Main Menu. This displays the Free Space Report Menu, Figure 5-4.



```
Free Space Report Menu

 0. Return to Main Menu
 1. Create free space report now
Enter Number> █
```

Figure 5-4. Free Space Report Menu

Enter **1** and then press <**Return**>. This creates the Free Space Report. An example is shown in Figure 5-5.



```
Database Free Space for database: P:ims
                       from node: eng207 on Tue Sep 19 13:32:00 1995

                                                                  Largest Free
Tablespace Name         Total Space  Used Space  Free Space  Cap.    Extent
----------------------  -----------  ----------  ----------  ---- ------------
HS_ARCHIVE                    5.00M       0.63M       4.36M   13%        4468K
HS_INDEXES                  300.00M     275.56M      24.44M   92%       21228K
HS_PDL                       20.01M       8.36M      11.65M   42%       11930K
HS_REPORTS                   10.00M       5.00M       5.00M   50%        5118K
HS_RESTORED                  10.00M       4.08M       5.92M   41%        6058K
INFORM_HS_RUNTIME           525.00M     473.77M      51.22M   90%       52454K
SYSTEM                      100.00M      29.68M      70.31M   30%       72002K


Log Capacity Summation by Object Type:

Log Type      Count   Capacity
--------      -----   --------
Numeric        5042     7951216
SPC             504    1480752
Message           1       6000
Other             1        200
```

Figure 5-5. Example, Free Space Report

5.6.4 Entry Tables Report

This function generates a report of all numeric logs, their current capacity, and other useful statistics. You can run this function either by choosing the Entry Tables report option from the Main Menu, or by entering a command in a terminal window command line.

If you run this function via the menu, you can modify these report settings before you run the report:

- log ID to process. You may choose to generate a report for all logs, just one log, or a range of logs. The default is for ALL LOGS.
- change the correction of bad logs. The default is NONE.
- change the line per page. The default is 24 lines per page.

If the above default settings are OK, it is best to run from the command line to allow output to be directed to a file. If you need to change a default setting, use the main menu.

NOTE

This function may take a couple hours to complete for a large database, and generates a high CPU load.

5.6.4.1 Running Entry Tables Report Function from Command Line

This function will generate a report for all logs and direct the report output to a specified file.

To run the report, enter:

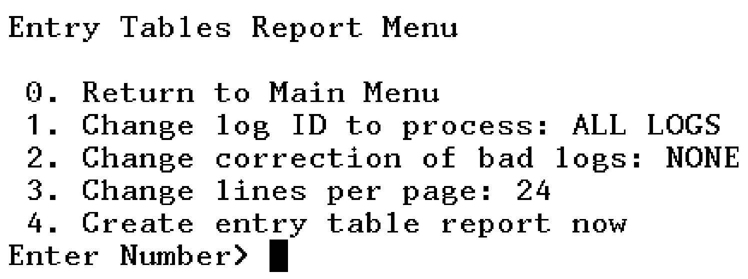
```
hsDBMaint -report > /tmp/filename.txt
```

where *filename* is the name of the file where the report output will be directed to.

The file is given a .txt extension. You can view the file using an editor such as vi.

5.6.4.2 Running Entry Tables Report Function from the Menu

To create an entry tables report using the main menu, choose **Entry Tables Report** from the Main Menu. This displays the Entry Tables Report Menu, Figure 5-6.



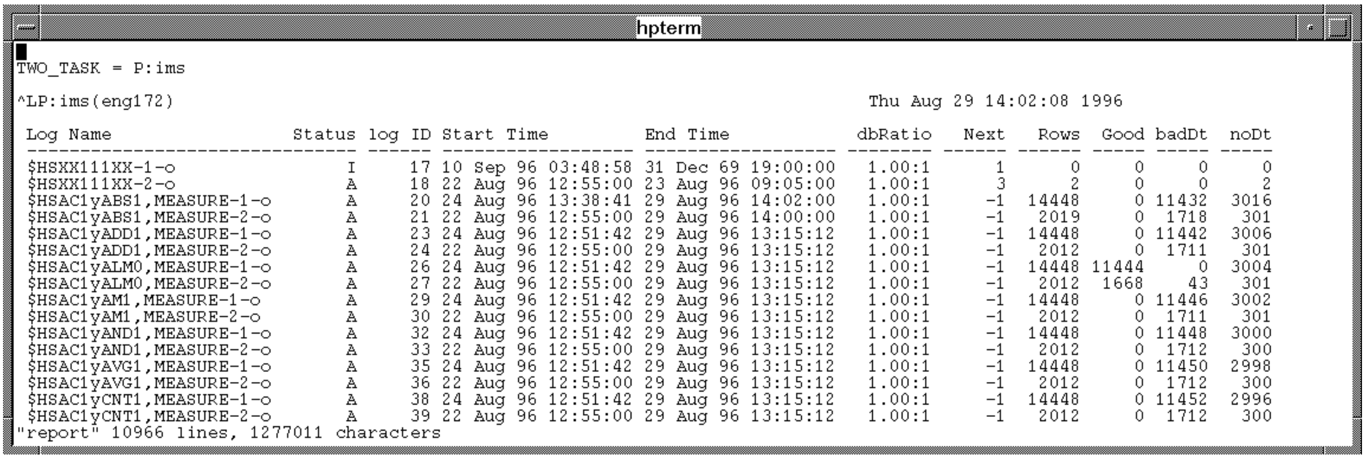
```
Entry Tables Report Menu

 0. Return to Main Menu
 1. Change log ID to process: ALL LOGS
 2. Change correction of bad logs: NONE
 3. Change lines per page: 24
 4. Create entry table report now
Enter Number> █
```

Figure 5-6. Entry Tables Report Menu

This menu shows you the current settings for log ID to process, correction of bad logs, and lines per page. You can change the current settings before you run the report, or you can return to the Main Menu without using any of these functions.

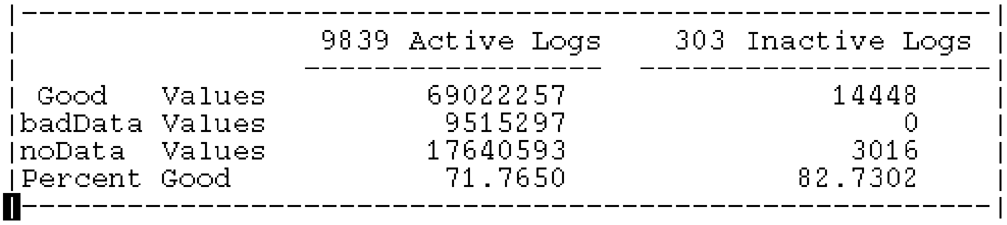
If you need to change any of the settings, choose the corresponding option from this menu and follow the dialog. When you finish with the dialog, you will return to the Entry Tables Report menu and the current settings will be changed according to the dialog. When the settings are correct, choose the Reset specified log(s) with current settings now option (3). This will generate the report which is displayed in the terminal window, Figure 5-7. A short summary is provided at the end of the report, Figure 5-8. When the report is finished, the Main Menu will be displayed.

If you need to return to the Main Menu without invoking the Entry Tables Report function, choose the Return to Main Menu option (0).



```
TWO_TASK = P:ims

^LP:ims(eng172)                                                                 Thu Aug 29 14:02:08 1996

 Log Name                 Status log ID Start Time         End Time            dbRatio   Next   Rows  Good badDt  noDt
 ------------------------------- ------ ------------------ ------------------ -------- ------ ------ ----- ----- -----
 $HSXX111XX-1-o                I     17 10 Sep 96 03:48:58 31 Dec 69 19:00:00   1.00:1      1      0     0     0     0
 $HSXX111XX-2-o                A     18 22 Aug 96 12:55:00 23 Aug 96 09:05:00   1.00:1      3      2     0     0     2
 $HSAC1yABS1,MEASURE-1-o       A     20 24 Aug 96 13:38:41 29 Aug 96 14:02:00   1.00:1     -1  14448     0 11432  3016
 $HSAC1yABS1,MEASURE-2-o       A     21 22 Aug 96 12:55:00 29 Aug 96 14:00:00   1.00:1     -1   2019     0  1718   301
 $HSAC1yADD1,MEASURE-1-o       A     23 24 Aug 96 12:51:42 29 Aug 96 13:15:12   1.00:1     -1  14448     0 11442  3006
 $HSAC1yADD1,MEASURE-2-o       A     24 22 Aug 96 12:55:00 29 Aug 96 13:15:12   1.00:1     -1   2012     0  1711   301
 $HSAC1yALM0,MEASURE-1-o       A     26 24 Aug 96 12:51:42 29 Aug 96 13:15:12   1.00:1     -1  14448 11444     0  3004
 $HSAC1yALM0,MEASURE-2-o       A     27 22 Aug 96 12:55:00 29 Aug 96 13:15:12   1.00:1     -1   2012  1668    43   301
 $HSAC1yAM1,MEASURE-1-o        A     29 24 Aug 96 12:51:42 29 Aug 96 13:15:12   1.00:1     -1  14448     0 11446  3002
 $HSAC1yAM1,MEASURE-2-o        A     30 22 Aug 96 12:55:00 29 Aug 96 13:15:12   1.00:1     -1   2012     0  1711   301
 $HSAC1yAND1,MEASURE-1-o       A     32 24 Aug 96 12:51:42 29 Aug 96 13:15:12   1.00:1     -1  14448     0 11448  3000
 $HSAC1yAND1,MEASURE-2-o       A     33 22 Aug 96 12:55:00 29 Aug 96 13:15:12   1.00:1     -1   2012     0  1712   300
 $HSAC1yAVG1,MEASURE-1-o       A     35 24 Aug 96 12:51:42 29 Aug 96 13:15:12   1.00:1     -1  14448     0 11450  2998
 $HSAC1yAVG1,MEASURE-2-o       A     36 22 Aug 96 12:55:00 29 Aug 96 13:15:12   1.00:1     -1   2012     0  1712   300
 $HSAC1yCNT1,MEASURE-1-o       A     38 24 Aug 96 12:51:42 29 Aug 96 13:15:12   1.00:1     -1  14448     0 11452  2996
 $HSAC1yCNT1,MEASURE-2-o       A     39 22 Aug 96 12:55:00 29 Aug 96 13:15:12   1.00:1     -1   2012     0  1712   300
"report" 10966 lines, 1277011 characters
```

Figure 5-7. Sample Entry Tables Report



```
|--------------------------------------------------------|
|                 9839 Active Logs     303 Inactive Logs |
|                ------------------  --------------------|
| Good   Values          69022257               14448    |
|badData Values           9515297                   0    |
|noData  Values          17640593                3016    |
|Percent Good             71.7650             82.7302    |
|--------------------------------------------------------|
```

Figure 5-8. Report Summary

5.6.5 Update Deadband Ratio On-line

Use this function to update deadband ratio and estimated log time period on-line for logs that use deadband.

Disk space is allocated to logs according to their log capacity, and not the amount of data actually stored. The log period can be adjusted on an individual log basis to keep log capacity at a reasonable size for each log. For all synchronous logs with or without data compaction, log capacity is calculated as follows:

log capacity = (log period/storage interval)

When you use compaction, the log period is effectively increased so that you store the same number of samples (as determined by the log capacity) over a longer period of time. Consider the following two cases. If your process is very unstable such that every sample is stored and there is no data compaction, the effective log period will be equal to the configured log period. If your process is perfectly stable (maximum data compaction, samples stored only at deadband storage interval), the effective log period is extended according to the following formula:

effective log period = (deadband storage interval/storage interval) * configured log period

In most cases, since the amount of process variation will not be at either extreme, the effective log period will fall somewhere in between the configured (minimum) and maximum value.

The effective log period calculated according to the above formula is stored in the EST_LOG_TIME_PER attribute. This attribute is updated periodically according to a schedule which repeats as often as every five days. If necessary, you can manually execute the update function via the hsDBMaint utility.

EST_LOG_TIME_PER is used by the Log Manager when it needs to determine which log to retrieve data from (seamless retrieval). For instance, consider an application with a composite log where the primary log stores data at 30 second intervals over a one-day time period, and the secondary log stores data at one-hour intervals over a one-week time period. Your data compaction ratio is 5:1 which extends the effective log period of the primary log to five days. if you make a request to retrieve data that is three days old, the log manager will know to get the data from the primary log rather than the secondary log, even through the configured log period of the primary log is only one day.

When EST_LOG_TIME_PER is updated, both internal History memory and the History database are updated with the new values of EST_LOG_TIME_PER and COMPACTION_RATIO. This will allow Log Managers access to the information and when the History node is restarted the information will also be available.

To update deadband ratio and estimated log time period, choose **Update Deadband Ratio On-line** from the Main Menu. This will generate a prompt that asks whether or not you want to continue. Press y (Yes) to continue.

This can take several hours for a large database.

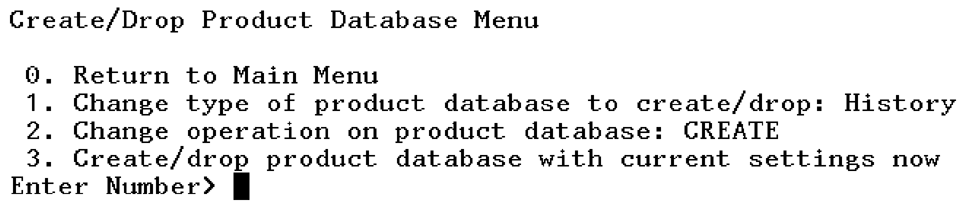
5.6.6 Create or Drop a Product Database

Use this function to create or drop (delete) a database. When you drop a database, all configuration data and stored log data is deleted. As an example, you may use this function while you are configuring your History database, and you determine that it would be better to delete your work up to this point and start from scratch. In this case you would drop the current database, and then create a new one.

CAUTION

> When you drop a database, all configuration data and log data is lost. If you need to drop a database after History has been running (database has live data), be sure to archive any data that you do not want to lose.

To create or drop a product database, choose **Create/Drop Product Database** from the Main Menu. This displays the Create/Drop Product Database menu, Figure 5-9.



```
Create/Drop Product Database Menu

 0. Return to Main Menu
 1. Change type of product database to create/drop: History
 2. Change operation on product database: CREATE
 3. Create/drop product database with current settings now
Enter Number> █
```

Figure 5-9. Create/Drop Product Database Menu

This menu shows you the current settings for type of database to create/drop, and the operation to perform (create or drop). You can change the current settings before you execute the create or drop operation, or you can return to the main menu without performing any of these functions.

If you need to change any of the settings, choose the corresponding option from this menu and follow the dialog. For type of database you can choose:

- History Database
- PDL Database
- SPC Database

For operation to perform you can choose:

- Create
- Drop

Creating History will automatically create SPC and/or PDL if those options are installed.

When you finish with the dialog, you will return to the Create/Drop Product Database menu and the current settings will be changed according to the dialog. When the settings are correct, choose the Create/drop product database with current settings now option. When the operation is finished, the Main Menu will be displayed again.

If you need to return to the Database Maintenance Program main menu without invoking any of these functions, choose the Return to Main Menu option (0).

5.6.7 How to Extend Tablespace for History Files

Extend the tablespace in your system when you require greater storage capacity than is provided by the default tablespace sizes as described in Section 3.2.2, Disk Capacity.

5.6.7.1 Guidelines for Extending Tablespace

Follow these guidelines to determine which tablespace(s) to extend and how much to extend them by:

- Determine how much more space you will require for additional log entries.

  Each Oracle-based numeric log entry occupies 45 bytes in the Inform_HS_RunTime tablespace. Multiply the number of additional entries by 45. This gives the additional space you need in bytes. For instance, if you need 100,000 additional log entries, then you must extend the Inform_HS_RunTime tablespace by 4.5 meg (100,000 * 45 = 4,500,000 bytes = 4.5 meg).

- When you extend the Inform_HS_RunTime tablespace, you must also extend the HS_Indexes tablespace by 80% of the amount that you extend the Inform_HS_RunTime tablespace by.

  For instance, if you add 4.5 meg to the Inform_HS_RunTime tablespace, you must extend the Inform_HS_Indexes tablespace by 3.375 meg (4.5 meg * 80% = 3.375 meg).

- Generally, you should not have to extend the HS_Restored tablespace. You can control the number of restored logs as described in Section 4.4.3.2, How to Archive Logs.

  If you need to extend the HS_Restored tablespace, use the method above to determine how much space you will require (45 bytes per log entry), and extend the HS_Indexes tablespace by 80% of the amount that you extend the HS_Restored tablespace by.

5.6.7.2 Procedure for Extending Tablespace

First glance at the current settings in the Tablespace Extension menu for which tablespace to extend, the size to extend by, and the directory to write data to (defaults are: INFORM_HS_RUNTIME, 50 Mb, and `/home/opt/advant/History/oracle` respectively).

If these settings are acceptable, follow this abbreviated procedure:

1. Choose **Extend a Tablespace** on the Main Menu.

   This displays the Tablespace Extension Menu, Figure 5-10.

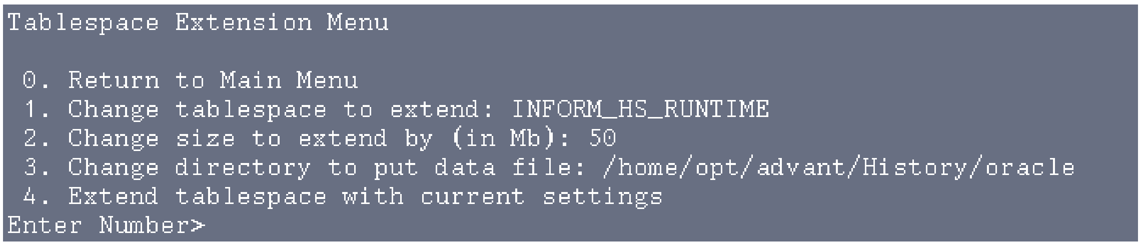
2. Enter **4** and then press <**Return**>.

   This extends the tablespace by the specified amount, and then returns you to the Tablespace Extension menu.

If one or more of these settings need to be changed, then follow this complete procedure:

1. Choose **Extend a Tablespace** on the Main Menu.

This displays the Tablespace Extension Menu, Figure 5-10.



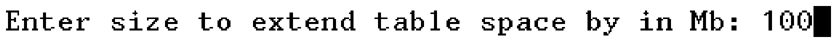
```
Tablespace Extension Menu

 0. Return to Main Menu
 1. Change tablespace to extend: INFORM_HS_RUNTIME
 2. Change size to extend by (in Mb): 50
 3. Change directory to put data file: /home/opt/advant/History/oracle
 4. Extend tablespace with current settings
Enter Number>
```

Figure 5-10. Tablespace Extension Menu

2. Specify the amount of space to add to the table (option 2). The default amount is 50 Mb as indicated in option 2. If this is acceptable, skip this step and go directly to step 3.
   a. To continue with this step and specify another amount, enter **2**, and then press <**Return**>.

      This displays a prompt to enter the size to extend table space by in Mb.

   b. Enter the size as shown in Figure 5-11, and then press <**Return**>.



```
Enter size to extend table space by in Mb: 100
```

Figure 5-11. Entering New Tablespace Size

      This returns to you to the Tablespace Extension menu with the new value indicated for option 2.

3. Specify the directory to write extended tablespace data to. The default directory is `/home/opt/advant/History/oracle` as indicated in option 3. If this is acceptable, skip this step and go directly to step 4.
   a. To continue with this step and specify another directory, enter **3**, and press <**Return**>.

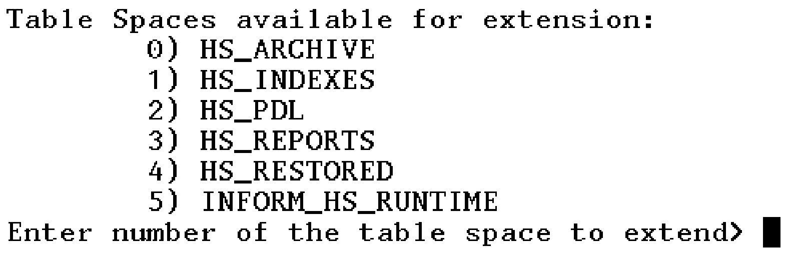
      This displays a prompt to enter the new data file directory.

   b. Enter the directory, and then press <**Return**>.

      This returns to you to the Tablespace Extension menu with the new directory indicated for option 3.

4. Specify the tablespace to extend. The default tablespace is INFORM_HS_RUNTIME as indicated in option 1. If this is acceptable, skip this step and go directly to step 5.
   a. To continue with this step and specify another tablespace, enter **1**, and press <**Return**>.

This displays the Table Spaces available for extension menu, Figure 5-12.



```
Table Spaces available for extension:
        0) HS_ARCHIVE
        1) HS_INDEXES
        2) HS_PDL
        3) HS_REPORTS
        4) HS_RESTORED
        5) INFORM_HS_RUNTIME
Enter number of the table space to extend> █
```

Figure 5-12. Table Spaces Available for Extension Menu

b. Enter the number corresponding to the tablespace you want to extend (for example 3 for HS_REPORTS), and then press <**Return**>.

   This returns to you to the Tablespace Extension menu with the new tablespace indicated for option 1.

At this point, your criteria for extending tablespace should be as you want it and you should be ready to actually extend the tablespace.

5. To extend tablespace with the current settings for size, directory, and tablespace, enter **4** and press <**Return**>.

   This extends the specified tablespace by the specified amount, and then returns you to the Tablespace Extension menu.

6. Repeat steps 1 to 5 to extend another tablespace, return to the Main Menu by entering **0** and then pressing <**Return**>.

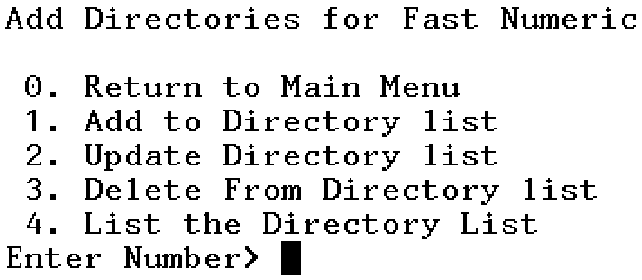
5.6.8 Add, Alter, Delete, or List Numeric Log Directories

This function is used to configure a directory structure for file-based storage of numeric logs. File-based storage is applicable for numeric logs whose Storage Type is configured to be TYPE1 or TYPE2.

This function provides basically the same functionality as the Directory Maintenance window. It is recommended that you use the Directory Maintenance window when History is running, since the user interface is better. If you need to configure your directory structure off-line, use the hsDBMaint function.

For guidelines on directory maintenance, refer to Section 3.6.11.1, Guidelines for Configuring File-based Storage.

To configure your directory structure, choose **Add/Alter/Delete/List NumericLog Directories** from the Main Menu. This displays the Add Directories for Fast Numeric menu, Figure 5-13.



```
Add Directories for Fast Numeric

 0. Return to Main Menu
 1. Add to Directory list
 2. Update Directory list
 3. Delete From Directory list
 4. List the Directory List
Enter Number> █
```

Figure 5-13. Add Directories For Fast Numeric Menu

This menu provides the means for you to add a directory, change the quota for a directory, delete a directory, or list all directories. If you need to return to the Database Maintenance Program main menu without invoking any of these functions, choose the Return to Main Menu option (0).

5.6.8.1 Add a Directory

To add a directory, choose the Add to Directory List option (1). This initiates a dialog for configuring a new directory. You will be prompted to specify a quota and path for the new directory. Follow the guidelines provided by the dialog. When you define the path, make sure the disk where the directory is being added has enough space for the directory. A directory with the name HSFF will be added under the path that you enter. For example, if you specify the path as `/disk3`, a new directory will be created as follows:

```
/disk3/HSFF
```

5.6.8.2 Update Directory List

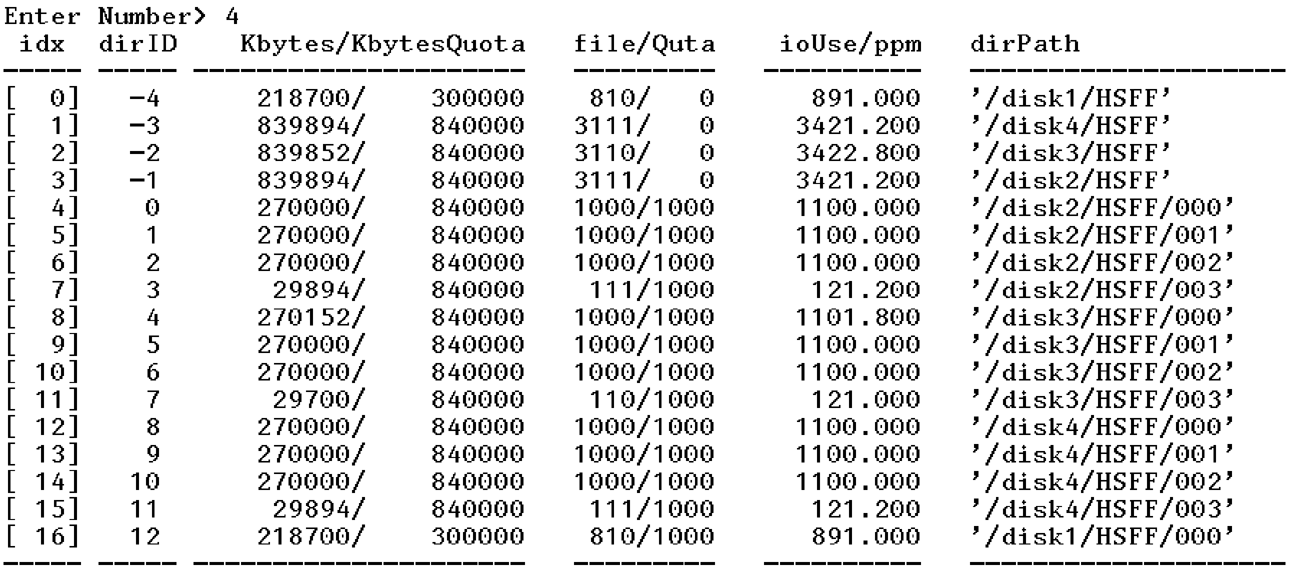
This function lets you change the quota for a specified directory. The dialog is the same as for adding a directory.

5.6.8.3 Delete From Directory List

This function lets you remove a directory from the list of directories that are available for file-based storage. Delete is not allowed if there are files in the directory.

5.6.8.4 List the Directory

This function provides a list of available directories, Figure 5-14. The information provided in this list is similar to the information provided in the Directory Maintenance window.



```
Enter Number> 4
 idx  dirID    Kbytes/KbytesQuota   file/Quta    ioUse/ppm   dirPath
----- ----- ---------------------   ---------   ----------   --------------------
[  0]   -4      218700/    300000    810/   0      891.000   '/disk1/HSFF'
[  1]   -3      839894/    840000   3111/   0     3421.200   '/disk4/HSFF'
[  2]   -2      839852/    840000   3110/   0     3422.800   '/disk3/HSFF'
[  3]   -1      839894/    840000   3111/   0     3421.200   '/disk2/HSFF'
[  4]    0      270000/    840000   1000/1000     1100.000   '/disk2/HSFF/000'
[  5]    1      270000/    840000   1000/1000     1100.000   '/disk2/HSFF/001'
[  6]    2      270000/    840000   1000/1000     1100.000   '/disk2/HSFF/002'
[  7]    3       29894/    840000    111/1000      121.200   '/disk2/HSFF/003'
[  8]    4      270152/    840000   1000/1000     1101.800   '/disk3/HSFF/000'
[  9]    5      270000/    840000   1000/1000     1100.000   '/disk3/HSFF/001'
[ 10]    6      270000/    840000   1000/1000     1100.000   '/disk3/HSFF/002'
[ 11]    7       29700/    840000    110/1000      121.000   '/disk3/HSFF/003'
[ 12]    8      270000/    840000   1000/1000     1100.000   '/disk4/HSFF/000'
[ 13]    9      270000/    840000   1000/1000     1100.000   '/disk4/HSFF/001'
[ 14]   10      270000/    840000   1000/1000     1100.000   '/disk4/HSFF/002'
[ 15]   11       29894/    840000    111/1000      121.200   '/disk4/HSFF/003'
[ 16]   12      218700/    300000    810/1000      891.000   '/disk1/HSFF/000'
----- ----- ---------------------   ---------   ----------   --------------------
```

Figure 5-14. Example Directory List

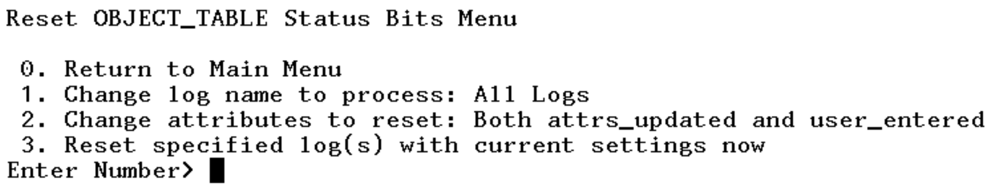
The columns in this list are as follows:

idx	This is a sequential index number
dirID	HSFF-level directories created under the disks that you add are assigned negative numbers, with -1 indicating the first disk added, and so on. Subdirectories under the HSFF-level directories are assigned positive numbers in the order they are created.
Kbytes	This indicates how many Kbytes have been used. For an HSFF-level directory, this is total usage for all subdirectories under it.
KbytesQuota	This indicates the quota allocated for the directory. For an HSFF-level directory, this is total quota for all subdirectories under it.
file	this indicates the number of files in the directory. For an HSFF-level directory, this represents the total number of files in all subdirectories under the HSFF directory.
Quota	This indicates the file quota. The File quota for HSFF directories is always zero (0) since files are always stored in the subdirectories. For subdirectories, the quota is always 1000.
ioUse/ppm	This is the average storage rate expressed in points per minute.
DirPath	This is the directory path.

5.6.9 Reset Object Status States

Use this function to reset *attrsUpdated* and *userEntered* bits. These bits determine whether the collection function for a log will update the data presentation attributes for numeric logs, or whether user-entered values will be used. If the *attrsUpdated* bit for a numeric log is set (1), when the log is activated, the collection function gets the data presentation attribute values from the data source. Any user-defined values will be overwritten. When the *userEntered* bit is set, it prevents the collection function from overwriting the user-defined values. At any given time, only one of these bits can be set for a numeric log. The Reset Object Status States function toggles the state for one or both bits.

To reset the status bits, choose **Reset the Object Status states** from the Main Menu. This displays the reset OBJECT_TABLE Status Bits menu, Figure 5-15.



```
Reset OBJECT_TABLE Status Bits Menu

 0. Return to Main Menu
 1. Change log name to process: All Logs
 2. Change attributes to reset: Both attrs_updated and user_entered
 3. Reset specified log(s) with current settings now
Enter Number> █
```

Figure 5-15. Reset OBJECT_TABLE Status Bits Menu

This menu shows you the current settings for logs to process, and attributes to reset. You can change the current settings before you execute the reset operation, or return to the Main Menu without performing any of these functions.

If you need to change any of the settings, choose the corresponding option from this menu and follow the dialog. For log name to process you can choose:

- all logs
- all logs in a specific composite log hierarchy
- just one specific log

For attributes to reset you can choose:

- both *attrsUpdated* and *userEntered* bits
- just *attrsUpdated*
- just *userEntered*

When the settings are correct, choose the Reset specified log(s) with current settings now option (3). When the function is finished, the Main Menu will be displayed.

If you need to return to the Database Maintenance Program main menu without invoking the Reset Object Status states function, choose the Return to Main Menu option (0).

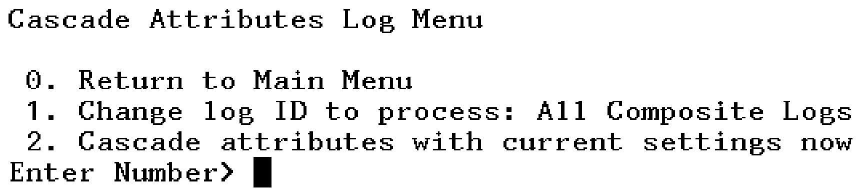
5.6.10 Cascade Attributes for Composite Logs

Use this function to propagate presentation, engineering units, and alarm attributes from the data source to secondary logs in the composite hierarchy (see Section 3.1.5, Presentation (Trend), Engineering Units, and Alarm Attributes). This is required for logs where attribute values have been entered manually.

To cascade attributes for a composite log:

1. Choose **Cascade Attributes for Composite Logs** on the Database Maintenance Program Main Menu.

   This displays the Cascade Attributes Log Menu, Figure 5-16.



```
Cascade Attributes Log Menu

 0. Return to Main Menu
 1. Change log ID to process: All Composite Logs
 2. Cascade attributes with current settings now
Enter Number> █
```

Figure 5-16. Cascade Attributes Log menu

2. Specify the log that you want to pass attributes to. The default is for all logs as indicated in option 1. If this is acceptable, skip this step and go directly to step 3. To specify a specific log, enter **1** and then press <**Return**>.

   This displays the following prompt:

   ```
   Do you want to do all composite logs? [yn]
   ```

   Enter **n**, and then press <**Return**>. This displays the following prompt:

   ```
   Enter the log ID of any log in the composite hierarchy>
   ```

   Enter the log name, and then press <**Return**>.

   This returns you to the Cascade Attributes Log Menu.

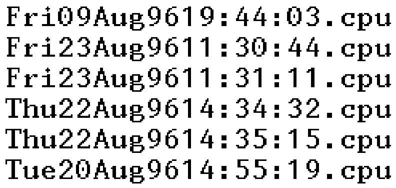
3. To cascade attributes with current setting for log, enter **2** and then press <**Return**>.

   This passes the applicable attributes from the data source to the specified log, and then returns you to the Cascade Attributes Log Menu.

4. To return to the Main Menu, enter **0** and then press <**Return**>.

5.6.11 Gather CPU Performance Statistics

Use this option to gather information on CPU usage to analyze performance problems. This function samples CPU usage at a specified sample rate. The performance test will continue until you stop it by pressing <**Ctrl - c**>, or until 1000 samples have been collected. The test results are stored in /usr/tmp. The filename for a performance test result is *date and time*.cpu as shown in Figure 5-17.



```
Fri09Aug9619:44:03.cpu
Fri23Aug9611:30:44.cpu
Fri23Aug9611:31:11.cpu
Thu22Aug9614:34:32.cpu
Thu22Aug9614:35:15.cpu
Tue20Aug9614:55:19.cpu
```

Figure 5-17. Example Performance Test Files Listed in /usr/tmp Directory

You can run a performance test, either using the Database Maintenance Program menu, or by invoking a command on the command line.

To use the menu, choose **Gather CPU performance stats** from the main menu. If you want to end the test before 1000 samples are collected, press <**Ctrl-C**>. This test uses the default sample time which is 15 minutes.

To invoke this function from the command line, enter:

```
hsDBMaint -perf -t n
```

where n is the sample time in seconds (for example 60 = 1 minute, 300 = 5 minutes)

To see a list of options for this function, use the following command:

```
hsDBMaint -perf -?
```

To retrieve the data, use the following command:

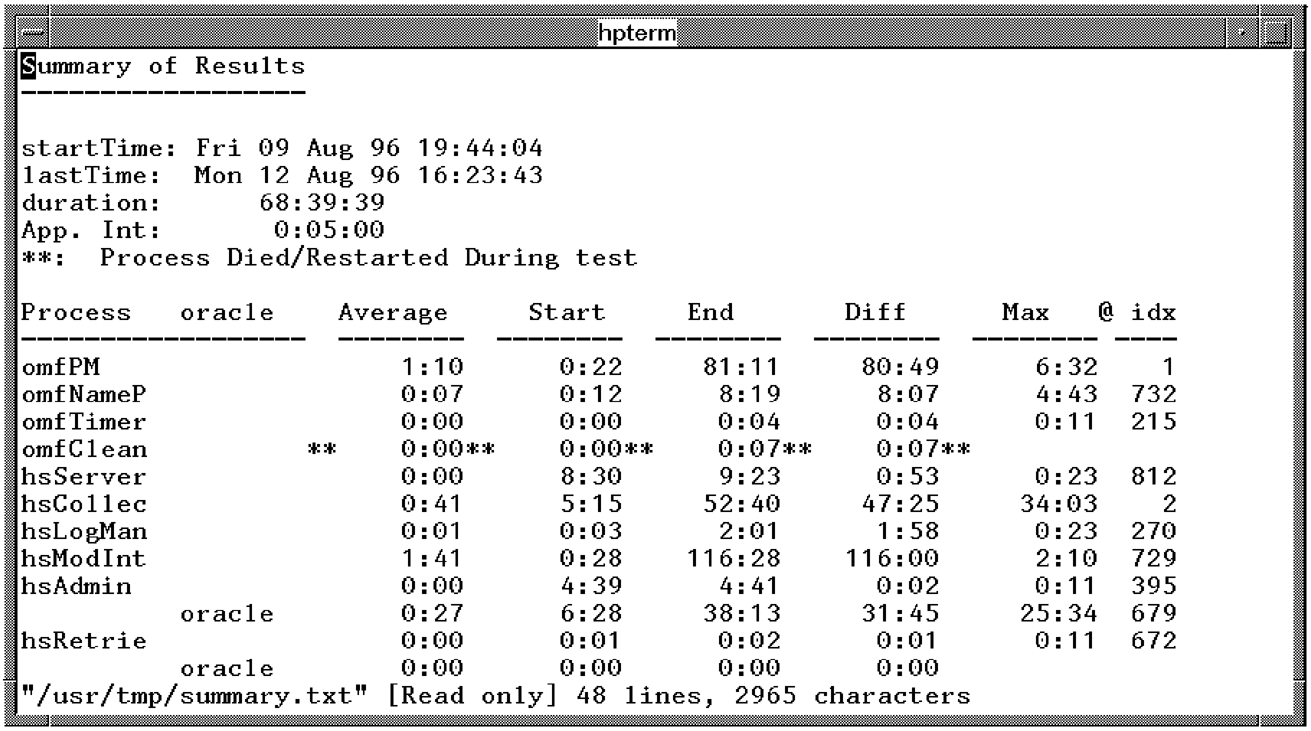
```
hsDBMaint -perf -r /usr/tmp/filename
```

where *filename* is the filename as shown in Figure 5-17. For example: Fri09Aug9619:44:03.cpu

Once you have retrieved the file, you can view it by entering the following command:

```
view /usr/tmp/summary.txt
```

An example is shown in Figure 5-18.



```
Summary of Results
------------------

startTime: Fri 09 Aug 96 19:44:04
lastTime:  Mon 12 Aug 96 16:23:43
duration:      68:39:39
App. Int:       0:05:00
**:  Process Died/Restarted During test

Process   oracle    Average     Start     End       Diff      Max    @ idx
------------------  --------  --------  --------  --------  -------- ----
omfPM                   1:10      0:22     81:11     80:49      6:32    1
omfNameP                0:07      0:12      8:19      8:07      4:43  732
omfTimer                0:00      0:00      0:04      0:04      0:11  215
omfClean          **    0:00**    0:00**    0:07**    0:07**
hsServer                0:00      8:30      9:23      0:53      0:23  812
hsCollec                0:41      5:15     52:40     47:25     34:03    2
hsLogMan                0:01      0:03      2:01      1:58      0:23  270
hsModInt                1:41      0:28    116:28    116:00      2:10  729
hsAdmin                 0:00      4:39      4:41      0:02      0:11  395
          oracle        0:27      6:28     38:13     31:45     25:34  679
hsRetrie                0:00      0:01      0:02      0:01      0:11  672
          oracle        0:00      0:00      0:00      0:00
"/usr/tmp/summary.txt" [Read only] 48 lines, 2965 characters
```

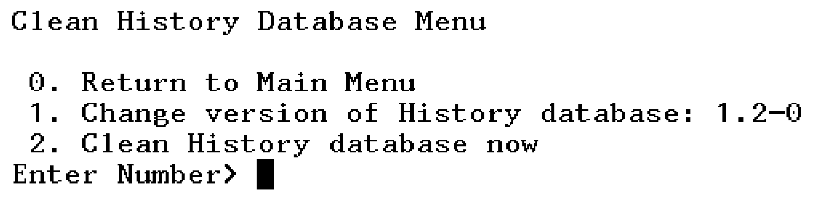
Figure 5-18. Example Performance Test Results

5.6.12 Clean History Database

Use this function to clean the History database. To clean the History database:

1. Choose **Clean History Database** on the Database Maintenance Program Main Menu.

   This displays the Clean History Database Menu, Figure 5-19.



```
Clean History Database Menu

 0. Return to Main Menu
 1. Change version of History database: 1.2-0
 2. Clean History database now
Enter Number> █
```

Figure 5-19. Clean History Database Menu

2. To clean the History database, enter **1** and press <**Return**>.

   This cleans up the database, and then returns you to the Clean History Database menu.

3. To return to the Main Menu, enter **0** and then press <**Return**>.

5.6.13 Restore or initialize History Logs

Use this function to:

- redefine missing tables for Oracle logs (storage type (Oracle) and missing files for file-based logs storage type (TYPE1 or TYPE2).
- initialize all log tables and log file essentials, and erasing all runtime data.

5.6.14 Stagger Collection and Storage

The collection and storage functions may cause spikes in CPU load when sample and storage rates are configured such that a large number of messages are written to disk at any one time. The Blocking Rate and Activation Time log attributes can be used to adjust collection and storage of TTD data, and storage of primary OCS data for a more even distribution of CPU load. These log attributes may be defined initially when you configure the History database. If you do not configure these attributes initially, or if you find that adjustments are required, the Stagger Collection function in the hsDBMaint utility provides a convenient means for adjusting these log attributes on-the-fly. This function must be run while History is running.

The Stagger Collection function provides two methods for distributing CPU load for collection and storage: Stagger and Phasing.

Stagger

Stagger is typically applied to a number of logs with the same alignment, storage interval, and sample interval, and where sample and storage interval are equal. Stagger assigns a blocking rate greater than the storage interval. The first Activation Time is modified to distribute the load. For example, consider an application with 120 logs with the following configuration:

- sample & storage interval = 30 seconds
- alignment = 12:00:00
- Blocking Rate = 150s

Without staggering, 120 messages with 5 entries each will be sent to storage every 2.5 minutes (150s). This will cause a spike in CPU usage. To distribute the load more evenly, you can use the Stagger Collection function to apply a blocking rate of 10m, and specify that 1/10 of the logs (12 messages with 20 entries each) be sent to storage every minute.

Phasing

Phasing is typically applied to a number of logs with the same alignment, storage interval, and sample interval, and the storage interval is significantly greater than the sample interval. For example, consider an application with 120 logs sampled at a 1-minute rate and an average calculated on a daily basis.

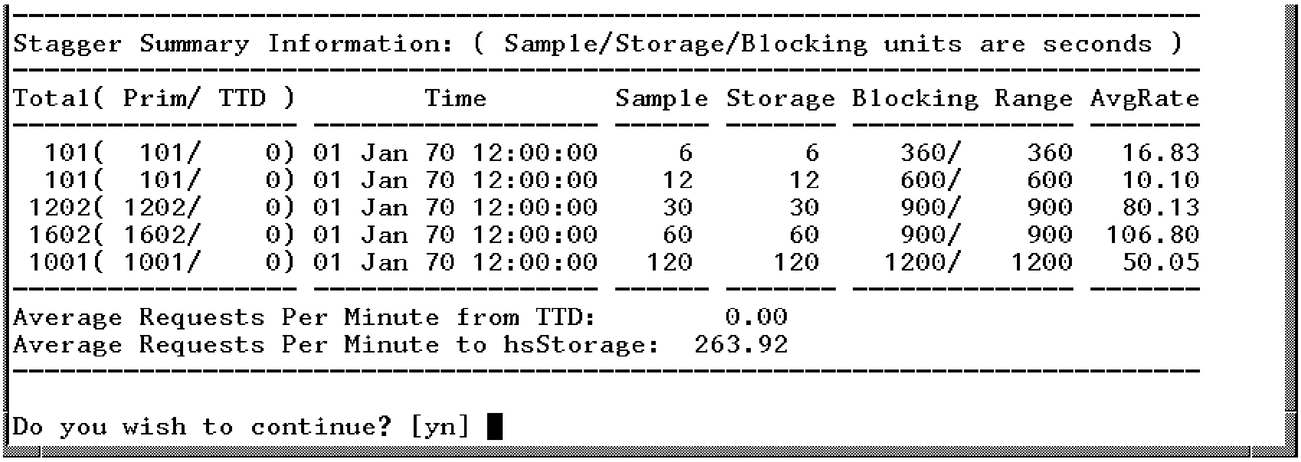
In this case, the 150s blocking rate would not prevent a CPU load spike at 00:00:00 (midnight). Phasing allows a set of logs to have slightly different blocking rates to prevent all logs from processing at the same time. For example, for a blocking rate of 30 minutes, phasing can be implemented by sub-dividing the 120 logs into groups and assigning different blocking rates to each group as follows:

- 40 logs w/ Blocking Rate = 29m
- 40 logs w/ Blocking Rate = 30m
- 40 logs w/ Blocking Rate = 31m

In this case a spike will only occur once every 26,979 minutes (29x30x31).

How to Use the Stagger Collection Function

To start the Stagger function, choose **Stagger Collection of data to improve performance** from the main menu. This displays a summary of the current collection/storage performance, Figure 5-20.



```
---------------------------------------------------------------------------
Stagger Summary Information: ( Sample/Storage/Blocking units are seconds )
---------------------------------------------------------------------------
Total( Prim/ TTD )        Time         Sample Storage Blocking Range AvgRate
------------------ ------------------ ------ ------- -------------- -------
  101(  101/    0) 01 Jan 70 12:00:00      6       6     360/    360   16.83
  101(  101/    0) 01 Jan 70 12:00:00     12      12     600/    600   10.10
 1202( 1202/    0) 01 Jan 70 12:00:00     30      30     900/    900   80.13
 1602( 1602/    0) 01 Jan 70 12:00:00     60      60     900/    900  106.80
 1001( 1001/    0) 01 Jan 70 12:00:00    120     120    1200/   1200   50.05
------------------ ------------------ ------ ------- -------------- -------
Average Requests Per Minute from TTD:        0.00
Average Requests Per Minute to hsStorage:  263.92
---------------------------------------------------------------------------

Do you wish to continue? [yn] █
```

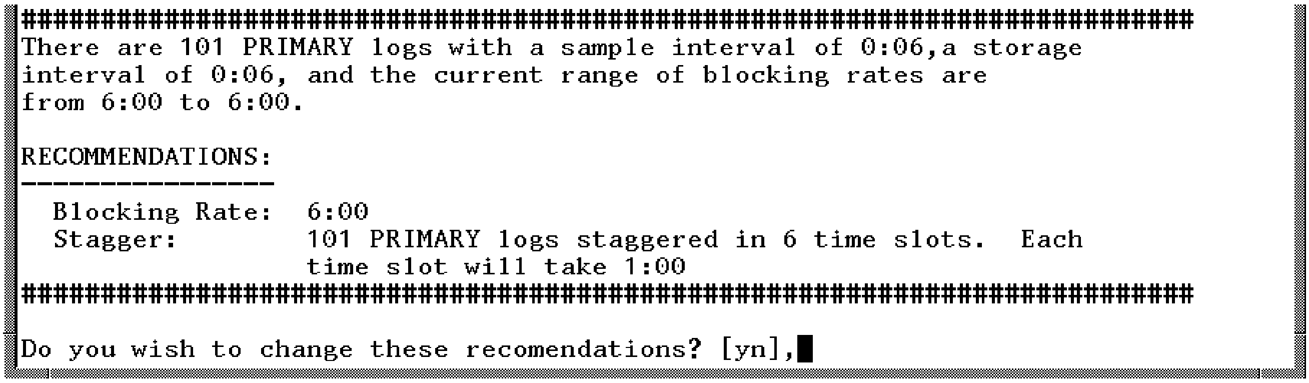
Figure 5-20. Summary of Collection/Storage Performance

The summary categorizes the logs by sample rate and storage rate. Each line of this summary provides the following information for a category of logs:

Total	Total number of logs in this category. This includes both cyclic subscriptions to OCS objects and periodic retrieval of TTD from controllers.
Prim	Number of logs in this category from cyclic subscriptions to OCS objects.
TTD	Number of logs in this category from periodic retrieval of TTD from controllers.
Time	First Activation time for logs in this category.
Sample	Sample rate for logs in this category (in seconds).
Storage	Storage rate for logs this category (in seconds).
Blocking Range	Shortest blocking rate for any logs in this category (in seconds)/ Longest blocking rate for any logs in this category (in seconds).
AvgRate	Calculated average number of requests per minute for this category based on number of logs, and sample, storage, and blocking rates. This equates to the average number of disk access transactions per minute for the system.

The prompt following the summary give you the opportunity to either exit hsDBMaint (by pressing n) or to continue with the Stagger function (by pressing y. Pressing n takes you out of the hsDBMaint function.

If you choose to continue with the Stagger function, recommendations to improve performance for logs in the first category are displayed, Figure 5-21.



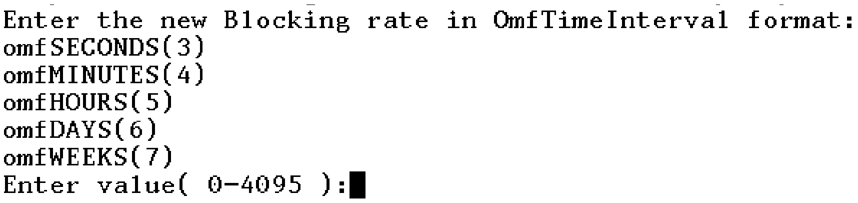
```
##############################################################################
There are 101 PRIMARY logs with a sample interval of 0:06,a storage
interval of 0:06, and the current range of blocking rates are
from 6:00 to 6:00.

RECOMMENDATIONS:
----------------
  Blocking Rate:  6:00
  Stagger:        101 PRIMARY logs staggered in 6 time slots.  Each
                  time slot will take 1:00
##############################################################################

Do you wish to change these recomendations? [yn],
```

Figure 5-21. Recommendations

The prompt following the recommendations gives you the opportunity to either accept the recommended blocking rate and stagger values for this category (by pressing n), or enter new values (by pressing y).

If you choose not to change the recommended values, you must respond to a prompt to confirm your selection, and then follow the same procedure for the next category (starting with recommendations for the next category).

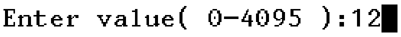
If you choose to change the values, instructions for entering a new blocking rate are displayed, Figure 5-22.



```
Enter the new Blocking rate in OmfTimeInterval format:
omfSECONDS(3)
omfMINUTES(4)
omfHOURS(5)
omfDAYS(6)
omfWEEKS(7)
Enter value( 0-4095 ):
```

Figure 5-22. Instructions for Changing Blocking Rate

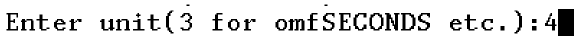
Enter a new blocking rate (without units) at the prompt. The range is 0 to 4095. For example, enter 12 as shown in Figure 5-23.



```
Enter value( 0-4095 ):12
```

Figure 5-23. Example, Entering Blocking Rate

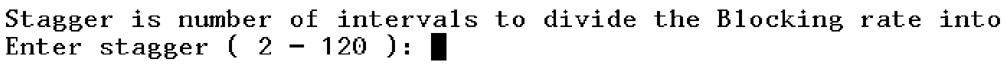
When the prompt for units is displayed, enter the units by entering the appropriate integer code: 3 = seconds, 4 = minutes, 5 = hours, 6 = days, 7 = weeks, Figure 5-24. In the example below, the integer code for minutes (4) is selected.



```
Enter unit(3 for omfSECONDS etc.):4█
```

Figure 5-24. Prompt for Units

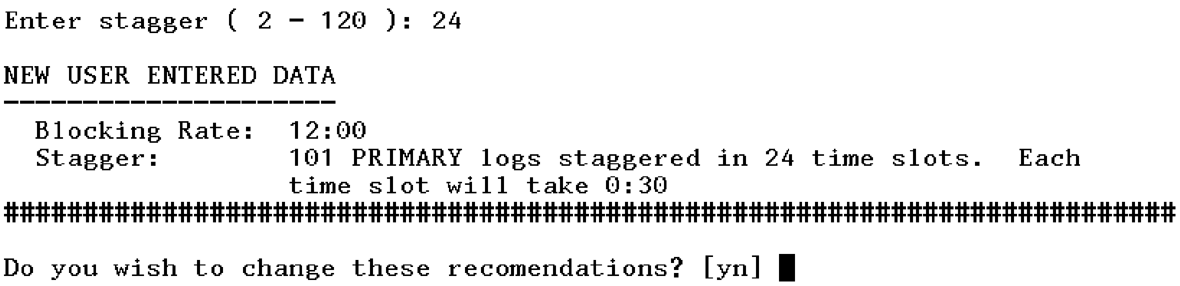
When the prompt for stagger is displayed, Figure 5-25, enter an integer value within the specified range. The range is based on the blocking rate that you specified.



```
Stagger is number of intervals to divide the Blocking rate into
Enter stagger ( 2 - 120 ): █
```

Figure 5-25. Prompt for Stagger

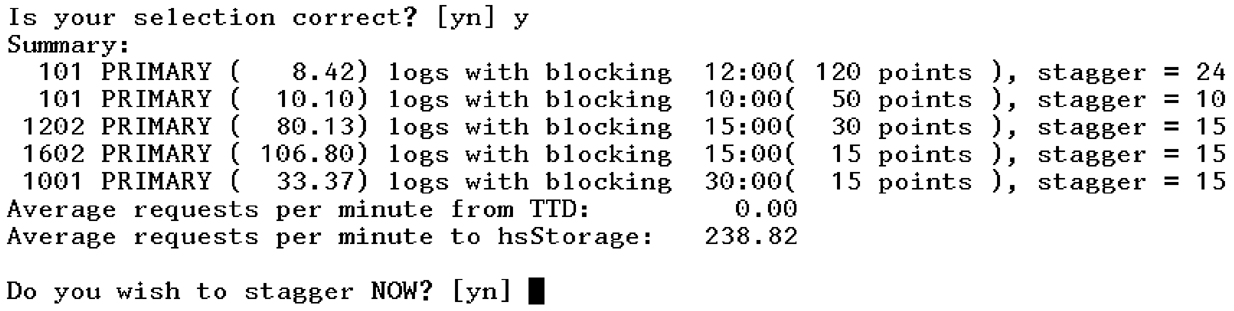
The stagger value specifies the number of divisions within the specified blocking rate that will be used to distribute the load. After you enter the stagger value, a summary of the changes you have made is displayed, Figure 5-26.



```
Enter stagger ( 2 - 120 ): 24

NEW USER ENTERED DATA
---------------------
  Blocking Rate:   12:00
  Stagger:         101 PRIMARY logs staggered in 24 time slots.  Each
                   time slot will take 0:30
###########################################################################

Do you wish to change these recomendations? [yn] █
```

Figure 5-26. Summary of Changes

The prompt following this summary gives you the opportunity to either accept these values (by pressing n), or change these values (by pressing y). In this case, the specified blocking rate is 12 minutes. The selected stagger divides each blocking interval into 24 30-second time slots. Thus at every 30-second time slot, This procedure must be repeated for each category. When you have completed this exercise for each category, a new summary will be displayed, and you will be prompted choose whether or not you want to apply the changes you have made, Figure 5-27.



```
Is your selection correct? [yn] y
Summary:
  101 PRIMARY (   8.42) logs with blocking  12:00( 120 points ), stagger = 24
  101 PRIMARY (  10.10) logs with blocking  10:00(  50 points ), stagger = 10
 1202 PRIMARY (  80.13) logs with blocking  15:00(  30 points ), stagger = 15
 1602 PRIMARY ( 106.80) logs with blocking  15:00(  15 points ), stagger = 15
 1001 PRIMARY (  33.37) logs with blocking  30:00(  15 points ), stagger = 15
Average requests per minute from TTD:         0.00
Average requests per minute to hsStorage:   238.82

Do you wish to stagger NOW? [yn] █
```

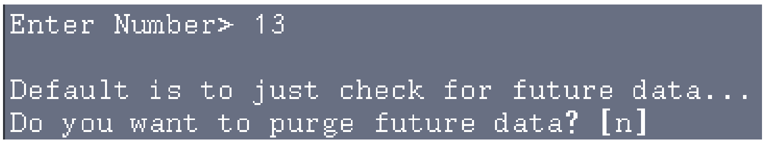
Figure 5-27. Revised Summary with Prompt to Apply Changes

Press `y` to apply the new stagger values, or `n` to ignore the changes you have made. The Advant Station must be restarted in order for the stagger changes to take effect.

5.6.15 Purge History Data

If, for some reason, your system time is set ahead this will cause History to collect data "in the future". For instance, this would occur if the time was set wrong via the IMSmenu. If this occurs, even if you fix the time, History will not collect any more data because it will not collect data that is older than the most current data collected. In this case you must use the Purge History Data function to delete all data that have time stamps in the future.

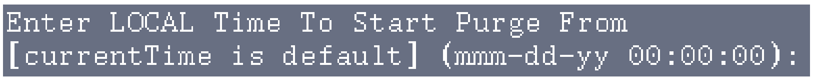
You can either use the -purge command line option, or choose **Purge History Data** from the Database Maintenance Program Main Menu. When you invoke this function you are prompted as to whether you want to actually purge the data, or just check which logs have future data, Figure 5-28. The default selection is to just check, and not purge. After you specify whether or not you want to purge, you are prompted for a time to compare data time stamps with, Figure 5-29. The default is the current time. When the function is finished, a table is displayed that indicates the number of logs checked, and the number of logs found to have future data, Figure 5-30. When History detects a time change of greater than 75 minutes (this is configurable via the environment variable HS_MAX_TIME_CHANGE), History will log an operator message stating that fact.



```
Enter Number> 13

Default is to just check for future data...
Do you want to purge future data? [n]
```

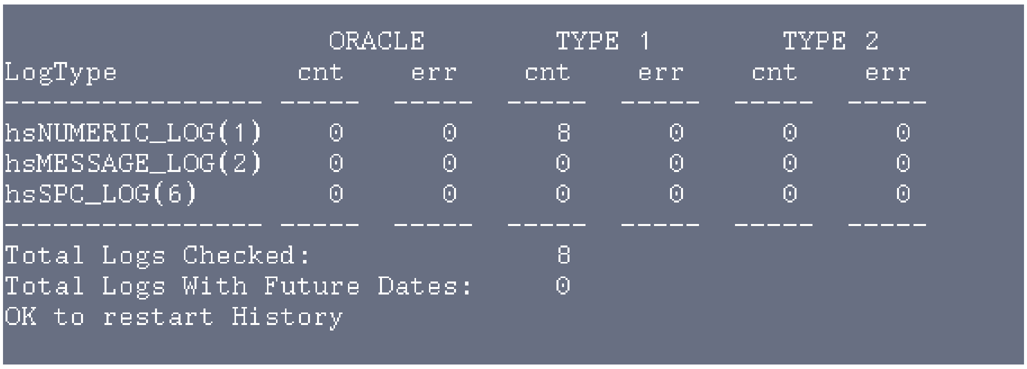
Figure 5-28. Prompt to Purge or Check

.



```
Enter LOCAL Time To Start Purge From
[currentTime is default] (mmm-dd-yy 00:00:00):
```

Figure 5-29. Prompt for Time



```
                    ORACLE        TYPE 1        TYPE 2
LogType            cnt    err    cnt    err    cnt    err
---------------- -----  -----  -----  -----  -----  -----
hsNUMERIC_LOG(1)    0      0      8      0      0      0
hsMESSAGE_LOG(2)    0      0      0      0      0      0
hsSPC_LOG(6)        0      0      0      0      0      0
---------------- -----  -----  -----  -----  -----  -----
Total Logs Checked:                  8
Total Logs With Future Dates:        0
OK to restart History
```

Figure 5-30. Purge Future Data Results

5.7 Backup and Restore

You can backup and restore the History database with the hsBAR utility. This includes the UNIX files that hold file-based logs, and Oracle tables that hold configuration data and any logs that are Oracle-based (Message logs and asynchronous numeric logs). hsBAR resides in the $HS_HOME/bin directory. To use this utility, you must be logged into AdvaInform as the ocsmgr user, or into AdvaCommand as operator. This utility is run from the command line at runtime.

The syntax for hsBAR is as follows:

```
$HS_HOME/bin/hsBAR -m bar [-d tape device name] [-f log file
name] [-p -n mount point] [-s -e path for export file]
```

where

`-m` is the operation mode. Operation mode is either:

`b` for backup the current history data to tape.

`r` for restore history data from a previously backed up tape.

`c` for check and determine the size of disk space needed for the temporary file holding the Oracle database. This is useful when backing up using the -s option or restoring a tape that was made with `-s` option.

`-d` <tape device name> is the tape device to use for backup. The default is: `/dev/rmt/0m`. When -s option is used, a remote tape device can be specified with the format machine:device.

NOTE

Always use the `m` device. Do not use the `mn` device.

`-f` <logfile name> is the full path and name of the file that will hold program output. The default for backup is: `/usr/tmp/historyBackup.log` The default for restore is: `/usr/tmp/historyRestore.log`

`-p` <-n mount point> is the path or new location for all restored flat files (ie. the files are restored based on the new mount point and not necessarily to the root). This is required primarily for conversions from HP-UX 9.x to HP-UX 10.x systems. If this option is not provided, all flat files are restored relative to root. If it is provided, either take the default or use the -n option to provide a mount point to use. The default mount point is `/home/opt/advant/History`

`-s` <-e path for export file> is the Single tape option where the exportfile is to be stored on disk prior to the backup operation. This eliminates the need for a separate tape to hold the Oracle export file by itself. If provided, either take the default or use the -e option to provide the full path name of where the export file is to be stored on disk. Make sure this is the LAST parameter supplied. If this option is not provided, the 2 tape method where the export and the numeric files are placed on separate tapes is used. The default export file path is `/home/opt/advant/History`

Operation Mode

The operation mode must be specified for hsBAR to work. The operation mode is specified by entering `-m` followed by either `b` for backup or `r` for restore. In backup mode, all data in the Oracle database owned by the Oracle History user is exported to tape, along with any files that have been created to hold numeric log entries.

In the restore mode, the History Oracle data is imported from tape into the Oracle database, and any files holding numeric log entries are copied from tape back onto the system. Restore can not be done while History is running.

Tape device Name

The default tape device is `/dev/rmt/0m`. If the target system has a tape device name that is different than the default, you can specify the tape device name using the -d option. To do so, enter -d followed by the full path and name. When you use the `-s` option, you can use -d to specify a remote tape device with the format machine:device. For instance:

```
$HS_HOME/bin/hsBAR -mb -d 999.999.99.99:/dev/rmt/0m -s
```

Log File

The log file is where all program progress messages and errors are printed to. The default log file is: `/usr/tmp/historyBackup.log` If you want to use a different file than the default, enter `-f` followed by the full path and name.

Mount Point

On restore, you can move the directory that file-based logs are kept in. For instance, `$HS_HOME/bin/hsBAR -mr -p -n /history` will restore files to the /history partition if it exists.

Single Tape Backup

This option lets you run a backup to tape without the need for operator intervention, as long as you are sure all data will fit on one tape.

Considerations:

- Make sure the tape device is available on the machine the backup or restore is to occur on.
- A backup requires at least two tapes (except when -s option is used). The first (n-1) holds the file-based log entries. Additional tapes may be required to hold file-based logs depending on the quantity. The last tape holds the Oracle export. The backup program prompts you when a new tape is required.
- When restoring, tapes must be restored in the same order that they were created, Also, the restore must be done using a tape that was used in a backup operation using hsBAR. No other backup utility tape formats will be recognized by the hsBAR utility.
- The log file should be checked after each invocation of hsBAR to make sure each operation completed successfully.

Examples:

In this example, hsBAR is invoked in the backup mode using the default tape device and log file.

```
$HS_HOME/bin/hsBAR -mb
```

In this example, hsBAR is invoked in the backup mode using the default log file, but specifying a different tape device.

```
$HS_HOME/bin/hsBAR -mb -d /dev/rmt/1m
```

In this example, hsBAR is invoked in the restore mode. Both the log file and tape device are specified. The specified tape device matches the device specified in the preceding backup example.

```
$HS_HOME/bin/hsBAR -mr -d /dev/rmt/1m
```

Appendix A History Resource Considerations

History places a load on the CPU, disk, and process memory resources in your History node. Data collection and storage place a fixed load on History node resources, based on your History database configuration. This load can only be adjusted by changing the database configuration. For instance you can adjust sample rates, deadband, blocking rates and so on. Data retrieval applications such as AdvaInform Reports, AdvaInform User API, and AdvaCommand trends, place a variable load that depends on the extent and frequency of those applications. For a detailed breakdown of these resources, refer to Section A.1, Resource Requirements Related to History Configuration and Section A.2, Resource Requirements Related to User Applications.

It is important to determine whether or not your History configuration and user applications will overload your History node resources. To analyze resource usage by your History configuration and user applications, refer to Section A.4, Determining Resource Usage for Your Application. This analysis will tell you where your configuration or user applications may overload your resources. Based on this analysis you can take any one or a combination of the following steps to make any necessary adjustments:

- Expand/upgrade the hardware for your History node (more disks, more memory, faster processor), or install a second node.
- Modify your History configuration.
- Modify your user applications.

A.1 Resource Requirements Related to History Configuration

CPU

CPU performance is related to capacity for History collection and storage functions. This can be adjusted by configuring sample rates, deadbands, and blocking rates.

Disk Space

Disk space resources are used by the History filesets that are loaded on disk when you install History, by Oracle tablespace for logs and archive database entries, and by file-based logs:

- The disk space required for the History filesets is fixed at 30 meg.
- Oracle tablespace is used by message logs, report logs, SPC logs, PDLs, asynchronous numeric logs, synchronous numeric logs with Oracle storage type, log configuration data, and archive database entries. The default Oracle tablespace capacities are defined in Section 3.2.2.1, Default Oracle Tablespace. You can extend Oracle tablespaces for runtime based on the per/entry requirements in Section 3.1.4.2, Disk Requirements for Oracle-based Logs and Section 3.1.4.3, Disk Requirements for File-based Numeric Logs.
- 200 MB file space for file-based numeric logs is automatically allocated. Additional disk space can be allocated. Memory requirements per file are described in Section 3.1.4.2, Disk Requirements for Oracle-based Logs and Section 3.1.4.3, Disk Requirements for File-based Numeric Logs.

Random Access Memory (RAM)

RAM resources are required for:

- HPUX resources. This includes:
  - Six Shared Memory IDs for:

    Runtime Shared Memory Size = (600+ total # of all logs) * 64 + (500 + # of numeric, report & message logs) * 184+ (100 + # of SPC logs) * 256 Bytes

    Other resources = 3.76 Meg.
- OMF
  - Object names = 500 (32 + # all logs -# numeric logs - # composite logs) Bytes
  - Advant OCS Object Data Sources = 1Kbyte per primary log minimum
- hsServer process = total # of History objects * 100 Bytes where History objects are Log Groups, Log Sets, Composite Log Typicals, Log Typicals, Composite Logs, and so on.
- hsCollection process = total # of all logs * 376 Bytes
- Overhead
  - History executables = 11 Meg.
  - Oracle Shadow Processes = 24 Meg. + (6 Meg. * # of archive devices)

A.2 Resource Requirements Related to User Applications

Load on resources for User applications comes from AdvaInform Reports, AdvaInform User API, AdvaInform Display, and AdvaCommand trends. These applications place a load on local and remote log managers:

- Requests from local Log Manager to local hsServer = 40Kbytes OMF shared memory per request for numeric log (500 numeric values maximum).
- Requests from remote Log Manager to local hsServer = 20Kbytes OMF shared memory per request for numeric log (500 numeric values maximum)

A.3 Other Considerations

History uses Oracle for everything except synchronous file-based logs. Performance for PDL, Report, Message, and SPC logs can be improved with additional dB Block Buffers. The default size is 1 Meg. Refer to the *Advant Station IMS User's Guide* for guidelines on how to expand this buffer space.

A.4 Determining Resource Usage for Your Application

Use the following questions as a guideline for determining how your History database configuration and user applications will load your resources. The affected resource is indicated in parenthesis:

- What type of Operator Station Graphics and Trending, Reports, and other user applications are expected:

  - How many operator stations will this History node service? (User Applications, CPU and Memory)
  - The frequency and type of applications using NUMLOGVAL? (User Applications, CPU and Memory)
  - Will AdvaInform Display be Installed? (User Applications, CPU and Memory)
  - Are any reports expecting certain types of data? (1 hour report returning 60 values, 1 for each minute in the hour) (History configuration, CPU, memory)

- Will you use SPC? If YES, you must consider the following:
  - SPC logs are still contained in ORACLE and require disk space. (Disk)
  - SPC logs generate alarms in systems with MOD 300 software.(History configuration, User Applications, CPU)
- Will you use PDL? If YES, you must consider the following:
  - Adjustment of HS_PDL tablespace to hold max number of PDLs.(Disk)
  - Log associations to numeric logs. The maximum log period should be greater than the expected duration of the batch to allow archival of all data and reporting to complete before the numeric log wraps.(History configuration, CPU memory)

- Will you use message logs? If YES, you must consider the following:
  - ORACLE tablespace is used for message logs.(Disk)
  - Capacities over 250,000 yield poor performance.(History configuration, CPU)
  - Will archiving be used for message logs (CPU, will need atrchive group).
- Will you use report logs? If YES, you must consider the following:
  - ORACLE tablespace (HS_REPORTS) is used to store reports in history. (Disk)
  - Will archiving be used for report logs (CPU, will need archive group).
- Will you use numeric logs? If YES, you must determine:
  - the number of source tags and their names. (Real Time Accelerator Board, CPU)
  - the rates to collect from tags. (100 logs at 30s, 200 logs at 1m, 1000 logs at 2m)(Real Time Accelerator Board, CPU)

    In MOD 300, sample rates for tags should be based on the loop processing rate configured for the tags in CCF (interaction of processing rate, processing phase, and base rate). A sample rate for a tag should never be faster than the tag's loop processing rate.

    In fact, in many cases, the sample rate can be slower than the loop processing rate. If you sample at a rate that is faster than the rate at which the process value changes, most of the sample data will be redundant, and will use CPU resources which could otherwise be used for additional logs. If a report requires 60 values for an hourly report, a 1m storage interval would make sense.
  - Can deadband be used. If YES, what is the expected compaction ratio?(5:1?) (History configuration, CPU, memory)
  - Will archive of data be necessary?
  - For all clients of history (numlogval, OS, reports, PDL), What log periods are required? (1 w, 2 w, 4w, etc.) (Disk)

    Log Period should be determined by the clients who view log data. The OS is the biggest concern here. Do the operators need 1 week or 4 weeks of on-line data.Batch applications are another consideration. The log associations in PDL require the log to have a log period at least 2 times the longest batch time to guarantee all the numeric data can be archived with the PDL. If all functionality for history data is clearly defined, a database satisfying 95% of the planned functionality can be created on the first try.

    In addition, if care is taken to only define what is needed (1 w logs instead of 1 year logs), overall system performance will be much better.
  - Do you need hierarchies?

- Will you use DUAL history? If YES, you must consider the following:
  - DUAL history allows history configurations to be distributed on multiple history nodes. A composite log with a primary log on two different history nodes is a DUAL history log. With a DUAL history log, each primary log collects data from the same data source independently. This means the data and timestamps will be similar but not equal for the two primary logs.
  - DUAL history allows clients of history (OS, REPORTS) to get data for logs when one of the two nodes is down.
  - DUAL history creates twice the load on controllers and the DCN.

  Having two nodes configured with DUAL history logs makes loading an issue. With DUAL history, both nodes use the same amount of RTA/OMF resources to collect data. However, the retrieval of data can be directed to one node more than the second node.

  In addition to the seamless algorithm applied for retrieval requests, the following criteria have been added to better handle selection of a log for a request by access name:
  - Uptime of node where log exists
  - local is chosen over remote (local should always be faster than remote)
  - sequence number of log. (all other conditions being equal, the lowest sequence numbered log will be used. Sequence number is the '-1-o' or '-2-o' attached to the generated log name.

  These additional sorting parameters will allow configurations to:
  - Evenly distribute retrieval load on two History nodes (on average) by creating half the logs with the first log created on Node A and the other half with the first log on Node B.
  - Favor one History node over the other by creating all the logs with the first log on NodeA (One history node may be faster and have more memory than the other.)

A.5 Temporary File Disk Space for History

History uses disk space for temporary files during runtime. This includes:

- Archive - 16 meg. per archive device.
- Inter-process communication - 5 meg.
- Collection snapshot files:
  - For Numeric, Message, and Report Logs: Minimum = # of Start Active Logs * 136, Maximum = 4 * (# of Start Active Logs * 136)
  - For SPC Logs: Minimum = # of Start Active Logs * 232, Maximum = 4 * (# of Start Active Logs * 232)

NOTE

Start Active logs are logs whose Log Start-up state is configured to be Active.

- Storage shutdown file - 2 to 5 Meg.

To assure that there is disk space to accommodate these temporary files, you must leave free space on disk 1, using the above numbers as guidelines. This free space is in addition to any other free space requirements that your History application may call for.

INDEX

A

- Access Name 3-22, 3-113
- Action Menu, Log Group Window 3-109, 4-40
- Action Menu, Log Set Window 3-111, 4-40
- Activate 3-109
- Activation 3-6, 3-33
- Activation Time 3-33
- AdvaBuild
  - Display Builder 1-2
  - Environment Builder 1-2
- AdvaInform
  - Basic Functions 1-2
  - History 1-1, 1-12
  - Reports 1-2, 1-22, 3-68
  - SPC 1-2, 1-23
  - SQL*Connect 1-22
  - User API 1-22
- AdvaInform User API 1-10
- Alarm 3-21
- Alarm Limit
  - High 3-38
  - High High 3-38
  - Low 3-38
  - Low Low 3-38
- Alignment 3-33
- API 1-10, 3-3, 3-34
- Archive
  - Add Archive Entries 1-24
  - Add Database Entries 4-33
  - Archive Database 4-27
  - Archive Database Window 4-27
  - Archive Database Window Menus 4-51
  - Archive Device Window 3-117
  - Archive Group 3-119
    - Window 3-118
  - Archive Logs Window 4-21
  - Archive Media Window Menus 4-45
  - Configuration 3-67
  - Delete Archive Entries 1-24, 4-31
  - Delete Restored Logs 1-24, 4-31
  - Device 4-20
  - Group 3-32, 4-21
  - How to Configure Archive Function 3-67
  - Initialize Media 1-24, 4-19
  - Initialize Media Window 4-19
  - Location 4-21 to 4-23
  - Log Class 4-21 to 4-23

A (continued)

  - Log Name 4-21 to 4-22
  - Log Type 4-21 to 4-23
  - Logs 1-24
  - Media 1-24, 4-33
  - Number of Logs 4-21 to 4-23
  - Restore Logs 1-24, 4-30
  - Retrieve by Group 4-24
  - Retrieve by Location 4-24
  - Status 4-21 to 4-23
  - View Archive Entries 1-24
- Archive Database
  - retrieve info 4-28
- Archive Group 4-22 to 4-23
- Archive Log Window File Menu 4-42
- Archive Log Window Menus 4-42
- attribute 1-10
- Attributes 1-18

B

- backup 5-40
- Bad Data Quality Limit 3-37
- Blocking Rate 3-6, 3-36

C

- Calculation Algorithm 3-5, 3-36
- Calculations 3-5, 3-12, 3-14, 3-36
  - Average 3-5
  - Instantaneous 3-5
  - Maximum 3-5
  - Minimum 3-5
  - Number of values 3-5
  - Standard deviation 3-5
  - Sum 3-5
  - Sum of squares 3-5
- Cascade Attributes 5-30
- Clean History Database 5-33
- Collection Mode 3-34, 3-108
  - API 3-3
- Collection Type 3-5, 3-34
- command 1-10
- Compaction 3-37
- Compaction Deadband 3-37
- Compaction Ratio 3-37
- COMPACTION_RATIO 5-22
- Composite Log

C (continued)

- Configure 3-48
- Configure using a Composite Log Typical 3-63
- Window 3-63, 3-112 to 3-113, 4-17

Composite Log Typical 3-54, 3-59
- Configure 3-59
- Window 3-103

Configuration Area 3-69
control network 1-10
controller 1-10
core system 1-10
CPU Performance 5-31

D

Data
- Archive 1-24
- Collection 1-19, 4-11
  - Activate 4-12
  - Deactivate 4-12
  - Synchronous 3-3
- Entry 3-17
- Presentation 1-21
- Retrieval 1-20
- Source 1-19, 3-35
- Storage
  - File_based 1-20
  - Oracle Table 1-20

Data Collection 3-3
- Storing Calculated Values 3-12
- Storing Calculated Values with Compaction 3-15
- Storing instantaneous values using compaction 3-11
- Storing Instantaneous values, no compaction 3-10

Data storage rate 3-39
Database
- create 5-23
- drop 5-23

Database maintenance 5-16
dbDump 3-74, 3-89
dbLoad 3-74, 3-90
DCS 1-10
Deactivate 3-109
Deadband 3-7, 3-37
Deadband Storage Interval 3-37
DI Value Logging 3-2
diagnostic messages 5-5
Directory Maintenance
- Window 3-122

Display
- Trend 1-21

D (continued)

Dual History 3-86
Dual history 1-16

E

Edit Menu, Composite Log Typical Window 3-105
Effective log period 3-8
Engineering Units 3-21, 3-38
- Maximum Value 3-38
- Minimum Value 3-38

Entry status 3-17
Entry Tables Report 5-20
Error Message 3-73
Error Messages 3-73, 5-2, 5-5
- HISTORY ACTIVATED 5-4
- HISTORY DEACTIVATED 5-4
- HISTORY INITIALIZATION FAILED 5-4
- HISTORY SHUTDOWN 5-4
- OPERATION OR ATTRIBUTE NOT IMPLEMENTED 5-4
- UNABLE TO INITIALIZE LOG STATES 5-5

ES 1-10
EST_LOG_TIME_PER 3-8, 5-22
Est_Log_Time_Per 3-37
Extending Tablespace 5-24
external node 1-11

F

Fault Finding and Repair 5-14
FCM 3-2
File Menu, Common 3-91
File Menu, PDL Window 4-58
File-based Storage 3-122
File-based storage 5-26
Free Space Report 5-19

G

Global Messages 3-96

H

Help Menu 3-93
History 1-1
- Builder 1-1, 3-1
- Logs 1-13
- Manager 3-44, 3-98
  - Status 3-97

History Builder Window Menus 3-94
History Configuration window 3-96

H (continued)
History Directories 3-65
History Manager 3-98
History Status 1-25, 3-97, 4-2
History Template 3-69
Host Computers 1-22
HP VUE 1-25
hsBAR 5-40
hsDBMaint 5-16

I
IMS 1-11
IMS menu 1-11
Initialize Media Window Menus 4-41
Interpolation 3-30

L
Location 3-108, 3-111
Log 1-12, 1-18
  Access Name 3-32
  activation/deactivation 4-11, 4-17
  archival 4-19, 4-21
  Attributes 1-18, 3-31, 3-51, 3-71
  Capacity 1-20, 3-6, 3-34
  Composite Log 1-14
  Composite Log Typical 1-17, 3-103
  Configuration 3-67
  Croup 3-46
  Description 3-32
  Directories 5-26
  Entry 1-13, 1-19
  Group 1-17, 3-32
  Hierarchy 1-14, 3-63
  Log Message Configuration 3-69
  Log Set 1-17, 3-32
  Log Set Configuration 3-65
  Log Type 1-18
  Log Typical 1-17, 3-106
  Name 3-22, 3-32
  Period 3-34
  Primary Log 1-14
  Production Data Log (PDL) 1-22
  Report Log Configuration 3-68
  restore 4-30
  Secondary Log 1-14
  Start-up State 3-33
  State 3-33
  Type
    Message 1-18

L (continued)
    Numeric 1-18
    Report 1-18
    SPC 1-18
Log Attributes 1-18
Log Entry
  Status 3-17
  Time Stamp 3-17
    Local Time 3-17
    Universal Time (UTC) 3-17
  Value 3-17
Log Group
  Window 3-107
Log group
  activation/deactivation 4-14
Log Groups 1-17, 4-14
Log Hierarchy 3-113
Log Menu, Composite Log Window 3-111
Log Period 3-6
Log Set 3-65, 4-12
  Window 3-110
Log set
  activation/deactivation 4-12
Log Set Configuration 3-100, 3-110
Log Set window 3-111
Log Size 3-40
Log Type 3-108
Log Typical
  Configure 3-54
  Window 3-106
Logs in Group 3-108
Logs in Set 3-111

M
Menus 3-91
Message box 3-73
Message Logs 3-69
messages 5-5

N
Numeric Log Attributes 3-34
NUMLOGVAL 3-26
  ENTRYSTATUS 3-26
  ENTRYVALUE 3-26
  Name 3-26
  OBJECTSTATUS 3-26
  Time 3-26

O
Object 1-11
    History 1-12, 3-69
    Process 1-12
    User 1-11
object types
    alarm and event 1-11
    history 1-11
    process 1-11
oerr 5-2
Oracle Trace Files 5-15

P
PDL 4-38
    archival 4-23
    delete 4-39
PDL Archive Configuration Window 3-128
Performance 3-39
Period 3-108
Presentation
    Maximum Value 3-38
    Minimum Value 3-38
Presentation Units 3-20
Process Object 1-11
Process Objects 1-12
Production Data Logs (PDL) 4-38

R
Report Log
    archival 4-22
Report Logs 3-68
Reset Object Status 5-29
restore 5-40
Retrieve by Location 4-24

S
Sample Interval 3-5, 3-35
Save As 3-64
Seamless retrieval 3-29
Set Name 3-111
Shut-down 2-1
SQL 1-12, 1-22
Start-up 2-1
State Set 3-38
Status Displays 5-15
Storage Interval 3-36, 3-108
Storage Type 3-35
Storage type 3-16
    ORACLE 3-16

S (continued)
    TYPE1 3-16
    TYPE2 3-16

T
Tablespace 3-40
    Extend 3-41
    Inform_HS_Runtime 5-24
Tag 3-2
TCL Billboard Message 1-4
Text File 3-74
Tools Menu, History Builder Window 3-95
Tools Menu, History Configuration Window 3-99
trend display
    AdvaCommand 1-21
    Multibus-based 1-12
Trend displays 1-21

U
Update deadband ratio 5-22
User API 1-10
USER_SUPPLIED 3-3, 3-34

V
Version 3-97
Volume ID 4-20
Volume label 4-20

ABB

3BUR 001 402 R0001

March 1994